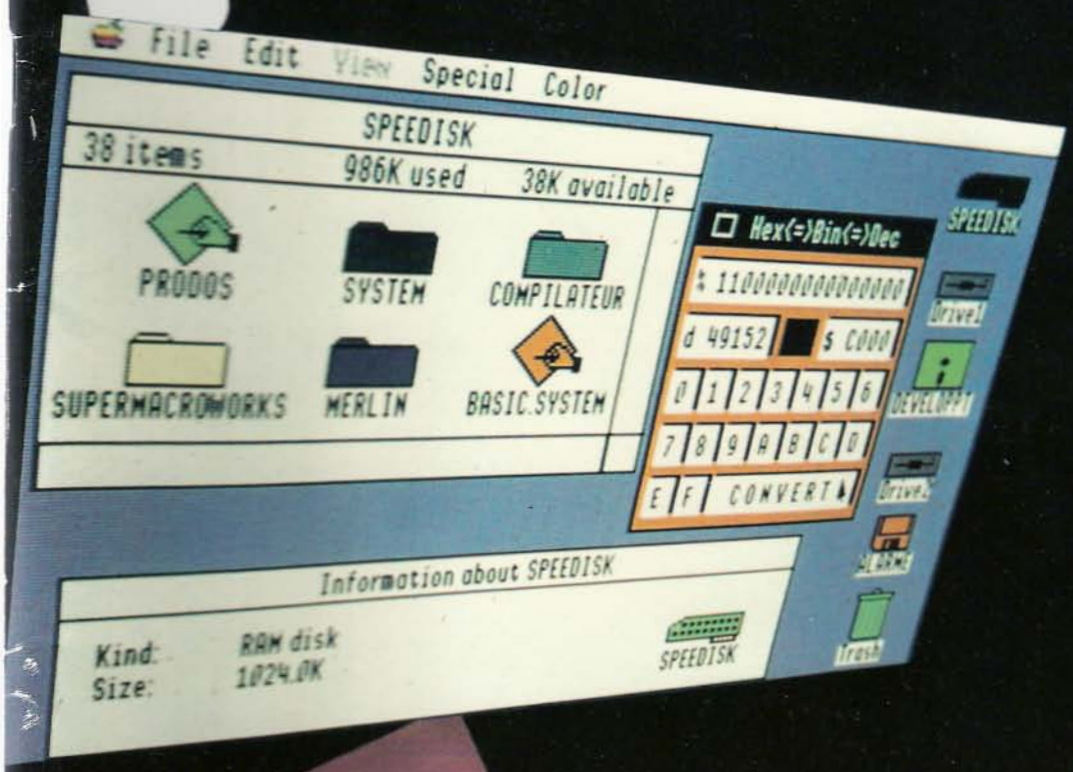
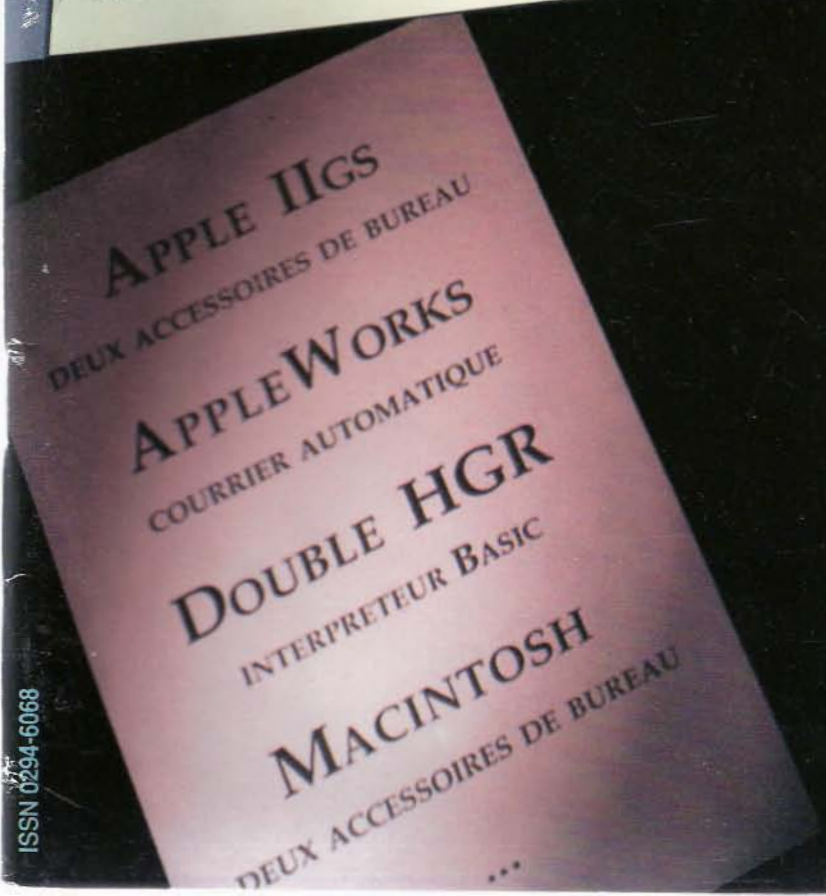
La revue francophone indépendante pour les utilisateurs des Apple ][+, //e, //e+, //c, IIGS™ et Macintosh™

# pom's




La carte Joker : Pom's-Assistance (1) 39 53 04 40 : Pom's sur Minitel



M 2366 - 35 - 45,00 F
3792366045005 00350
ISSN 0294-6068

# Jamais un outil aussi *rapide* n'a été aussi *simple*

Compiler un programme Basic
avec le BON OUTIL, c'est :

- Transformer le Basic en langage machine, sous ProDOS,
- Accélérer de 2 à 20 fois l'exécution des programmes,
- Générer un code plus compact que le Basic original,
- Compiler instantanément, lors du RUN, ou sur disque,
- Utiliser les cartes d'extension-mémoire,
- Gagner en confidentialité : le code est hermétique,
- Gagner en efficacité...

Compilateur, documentations française et américaine
750,00 F TTC, (abonnés 675,00 F), frais de port 20,00 F.
Pom's – Éditions MEV – 12, rue d'Anjou – 78000 Versailles
(commande par Minitel : (1) 39 53 04 40)
Banc d'essai, Pom' 35

Compilateur Beagle™

**Numéro 35**
**mars-avril 1988**


**Éditorial**
*Hervé Thiriez* — Page 7

**Multi-tâches : Share**
*Alain Raynaud* — Page 8

**un Classic Desk Accessory : Alarme**
*Sylvie Gallet* — Page 13

**Courrier automatique**
*Dimitri Geystor* — Page 21

**ProDOS : Petit guide de l'utilisateur**
*Pierre Demblon* — Page 25

**un New Desk Accessory : Converter**
*Vincent & Bernard Toméno* — Page 31

**Compilateur Beagle** — Page 35

**Big U de Beagle** — Page 37

**un accessoire de bureau Corner light**
*Alain Bohec* — Page 39

**En vitrine** — Page 44

**un accessoire de bureau Mini-montre**
*Alain Bohec* — Page 46

**Double-Graphics**
*Pascal Cantot* — Page 48

**Apple // for ever :**
*Jean-Yves Bourdin* — Page 57

**Courrier des Lecteurs** — Page 66

**Petites annonces** — Page 74

**Mots croisés**
*Joëlle Piard* — Page 74

Les annonceurs ; Apple : pages 4 et 5 ; LOGMA S.A. : page 11 : Softworld : page 13.



Éditions MEV - 12, rue d'Anjou - 78000 Versailles. Tél. : (1) 39 51 24 43. Directeur de la publication : Hervé Thiriez

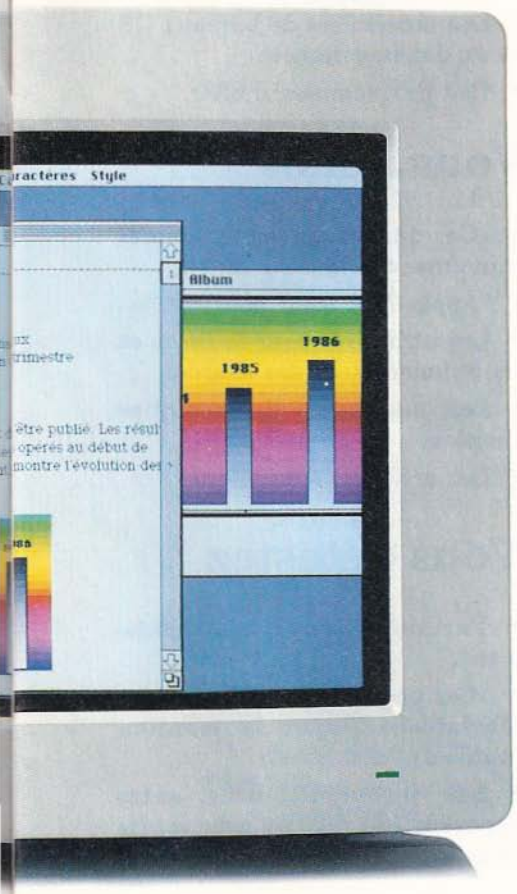
Imprimé en France. N° d'impression 785399 — Dépôt légal : mars 1988.

# Il a fallu d'abord réinventer des outils très simples...

Couper/coller, pour vous c'est naturel.
Pour un ordinateur c'est inhumain.
A tel point que certains ordinateurs préfèrent vous demander de changer vos habitudes.
Pour eux, c'est sans doute un point de détail mais pas pour Apple.
Plus votre façon de travailler est naturelle et intuitive, moins votre raisonnement sera encrassé de détails encombrants.

Macintosh était déjà l'outil idéal pour manipuler les idées complexes avec la même facilité que s'il s'agissait d'objets.
Macintosh II va encore plus loin, en y ajoutant ses capacités techniques incomparables : sa mémoire colossale, sa rapidité de calcul impressionnante, ses possibilités d'ouverture vers n'importe quel système et ses 16 millions de couleurs.

# pour que vous puissiez aujourd'hui manier des idées complexes.




Vous aviez déjà été ébloui par More (un logiciel d'organisation d'idées) sur Macintosh ; regardez-le encore, vous n'avez rien vu.

En couleur et à une vitesse incroyable, la hiérarchisation des idées et l'organisation des concepts deviennent si claires que Macintosh II semble sortir tout droit d'un album de Mandrake le Magicien.

Apple

# Macintosh II

C.L.M. & BBDO.

Apple, le logo Apple et Macintosh sont des marques déposées de Apple Computer Inc. Cricket Draw et Cricket Graph sont distribués par Alpha Systèmes. More est une marque déposée de A.C.I.

# Sondage : 130 % d'ordinateurs

**Voici donc le résultat de notre dernier sondage. Plus encore que les chiffres bruts qui se prêtent bien à la statistique, les commentaires qui accompagnent les bulletins (parfois plusieurs pages) nous sont un élément de réflexion et d'orientation indispensable.**

## Quel équipement !

Il est bien clair que pour le lecteur de Pom's, acheter un Apple IIGS ou un Macintosh ne signifie pas oublier sa première machine. Pour **100** lecteurs, on compte **130** ordinateurs !
Un lecteur sur six possède deux Apple ou plus. Dans la plupart des cas, on retrouve chez ceux-ci :

- un //e (ou c) et un IIGS,
- un Apple // et un Mac.

Autrement dit, on n'a pas revendu l'ancêtre lors de l'achat de la machine plus puissante.
Pour l'équipe de Pom's, voilà une explication du besoin de communication inter-machines fréquemment évoqué.

## //e, //c, IIGS : Match nul

Voici la répartition des matériels :

| | |
|---|---|
| I+ | 5,4% |
| //e | 33,6% |
| //c | 32,9% |
| IIGS | 34,2% |
| Mac 512 | 4,0% |
| Mac Plus | 14,8% |
| Mac SE | 5,4% |

Pour 100 lecteurs, il y a donc :

- **106** Apple // et
- **24** Macintosh.

Notons qu'il y a aujourd'hui une répartition équitable des //e, //c, IIGS.

Vous êtes **87,3%** à avoir une imprimante, essentiellement des ImageWriter/DMP.

Sur 100 imprimantes, il y a :

- **80,8%** d'ImageWriter/DMP et
- **19,2%** autres.

## Communication

Pour 100 lecteurs, nous avons trouvé :

- **49,0** Minitels et
- **23,5** modems

Même si quelques lecteurs ont Minitel ET modem, il faut compter que la majorité d'entre-vous (70%) a ouvert son ordinateur au monde extérieur. Grâce à Pom's ?

## Pom's et vous

Les rapports entre vous et Pom's d'une part, vous et votre ordinateur de l'autre, sont particulièrement stables.
Beaucoup de Basic, beaucoup d'assembleur, de nombreuses heures devant la machine.
Vous avez lu quasiment tous les Pom's et bien que notant votre revue sans complaisance, le niveau général atteint la plupart du temps 5 sur 5. Merci pour les auteurs.

## Vous réclamez

- Plus encore d'"Apple // for ever", rubrique particulièrement appréciée,
- Des routines 'Son' pour le GS,
- Des explications claires (nous nous y employons, croyez-le),
- Plus d'utilitaires de communication (avec les huit derniers numéros, nous avons déjà un ensemble cohérent, mais les idées ne manquent pas),
- Des accessoires de bureaux GS (deux dans ce numéro),
- Des programmes d'EAO.

## Vous aimez

- «Ce qui concerne ma machine/ma configuration» (!),
- "Apple // for ever" bien sûr,
- Les utilitaires pour le Basic et les commentaires des sources,
- Les montages électroniques simples,
- Les articles pédagogiques.

## Vous détestez

- Parution bimestrielle seulement,
- «Les pages Apple II/Les pages Macintosh» (rayer la mention inutile !),
- Les incompatibilités entre appareils (du type //c avec souris en port 7).

## Les heureux

Voici donc la liste de 10 lecteurs qui bénéficieront d'un abonnement :

**G. Covarelli** de Nice
**F. Muller** de Sélestat
**M. Rzewuski** de Cagnes sur Mer
**M. Gayrard** d'Antony
**A. Druyer** de Chalons sur Marne
**G. Rost** de Cergy
**M. Schwartz** de Lunéville
**R. Largeron** de Seyssinet
**M. Marquis** de Moelan
**P. Astier** de Paris

**Ont collaboré à ce numéro**
Alexandre Avrane – Jean-Luc Bazanegue
Alain Bohec - Jean-Yves Bourdin
Pascal Cantot - Pierre Demblon
Alexandre Duback – Sylvie Gallet
Dimitri Geystor - Olivier Herz
Gérard Michel – Christian Piard
Joëlle Piard – Alain Raynaud
Hervé Thiriez - Vincent & Bernard Toméno

**Directeur de la publication**
**rédacteur en chef**
Hervé Thiriez

**Rédacteurs**
Alexandre Avrane – Olivier Herz

**Siège social**
Éditions MEV – 12, rue d'Anjou
78000 Versailles – ☎ (1) 39.51.24.43

**Publicité**
Éditions MEV

**Diffusion**
N.M.P.P.

**Impression**
Berger-Levrault
18, rue des Glacis
54000 Nancy
☎ 83.35.61.44

**Photos p. 1, 2, 73, 75, 76**
CP & JLB

**Photogravure**
Graphotec
21, chemin de la Tour
92350 Le Plessis-Robinson
☎ (1) 46.30.44.49

Pom's est une revue indépendante non rattachée à Apple Computer, Inc. ni à Apple Computer France S.A.R.L.
Apple, le logo Apple, Mac et le logo Macintosh sont des marques déposées d'Apple Computer, Inc.
IBM est une marque déposée de International Business Machine.
PC et AT sont des marques déposées de la Société IBM.

©Éditions MEV 1988
Toute reproduction intégrale ou partielle, effectuée par quelque procédé que ce soit, sans l'accord écrit d'Éditions MEV, constitue une contrefaçon.
Loi du 11 mars 1957, articles 425 et suivants du Code Pénal.
Droits de traduction, de reproduction et d'adaptation réservés pour tous pays.

# *Éditorial*

## Une revue, un service et...

**Comme nous le laissions entendre dans notre précédent numéro, il nous fallait envisager des solutions pour renforcer et affiner la qualité du service Pom's. En effet, bien plus qu'une revue consacrée à la programmation, Pom's devient un véritable service ; même si nous tenons à rester dans le cadre des informations publiées dans la revue, comment ne pas répondre à une question concernant AppleWorks par exemple ?**
**Et comment satisfaire le souhait légitime d'un abonné de voir la confiance qu'il nous témoigne — souvent depuis le numéro un — prise en compte pour une assistance plus prompte, plus efficace ?**

## ... la carte 'joker'

**Si vous êtes abonné à Pom's, vous recevrez prochainement une carte — format carte de crédit, pratique et toujours sous la main — précisant un code personnel d'accès privilégié à notre 'hot line' ; téléphonique au (1) 39 51 24 43 ou télématique au (1) 39 53 04 40. Chaque carte est valide jusqu'à parution d'un numéro non compris dans l'abonnement.**

**Tout sera mis en œuvre pour apporter la solution au problème... quand bien même celui-ci sortirait quelque peu de la stricte application de nos programmes.**

**Autre 'plus' de notre carte — baptisée 'Joker' du fait de son aspect essentiellement dépannage —, des tarifs préférentiels précisés au fil des numéros (dès aujourd'hui 10 % sur l'excellent compilateur de Beagle, Pom_Link 3.1, SuperMacroWorks, Big U ou encore le détecteur de sonnerie).**

---

**Le serveur publié dans le numéro 34 a fait bien des petits : vous êtes déjà nombreux à nous avoir présenté vos réalisations, félicitations. Continuez, nous retiendrons certaines d'entre-elles.**
**Avez-vous croqué le module commande sur notre serveur ? Il s'écoule parfois moins d'une heure entre votre ordre et l'expédition de votre disquette... Vive le Minitel !**

**Hervé Thiriez**

---

*En couverture, le 'finder' de l'AppleIIGS de Dan Oliver & Cheryl Ewy livré avec le nouveau Système. Sur le bureau électronique, l'accessoire de bureau de Vincent & Bernard Toméno : Converter.*

# Share

# Un Apple, deux programmes

Alain Raynaud

Jusqu'à présent, votre Apple // ne savait exécuter qu'un seul programme à la fois. Or, il serait souvent bien agréable de pouvoir disposer d'un système, même rudimentaire, qui permette de faire tourner une 'tâche de fond' pendant que le travail principal s'exécute. L'accessoire de bureau sur le Macintosh est déjà un travail en multi-tâches, l'accessoire menant sa vie indépendamment des applications qui passent...

Cette lacune de l'Apple // va être comblée avec la routine SHARE qui s'installe sous ProDOS comme une commande externe. Une carte souris — déjà intégrée sur les //c — est nécessaire afin de générer les interruptions indispensables (désolé, mais la routine proposée ici ne fonctionne pas sur le IIGS).

## Cohabitez !

Tel est le mot d'ordre à donner aux différents programmes qui désirent s'exécuter simultanément en mémoire. Il n'est en effet pas question de pouvoir gérer les éventuels conflits de partage des zones mémoire ou des périphériques d'entrée-sortie.

## Utilisation

Le programme se lance par :

- *SHARE* ou

*BRUN SHARE*

Trois nouvelles commandes ProDOS sont alors disponibles :

**SHARE, Axxxx, Lyy** lance l'exécution en mode partagé du programme situé à l'adresse xxxx (indiquée en décimal ou en hexadécimal), en lui accordant yy% du temps machine disponible ; yy doit être choisi parmi les valeurs 25, 33, 50, 66 ou 75. Simultanément SHARE continue l'exécution du programme courant (en mode direct, il s'agit des interpréteurs ProDOS/Applesoft).

**NOSHARE** arrête l'exécution du programme lancé précédemment, et le mode multi-tâches. Cette conunande doit être *impérativement* lancée avant de soumettre une nouvelle commande *SHARE* pour faire exécuter un autre programme.

**DOSHARE** relance l'exécution en mode partagé du programme arrêté par *NOSHARE*, à partir de l'instruction qui suit la dernière exécutée.

## Exemple

Assemblez le petit programme TEST.S :

```
DEBUT  LDX £$80
LOOP   STX $C030
       DEX
       BPL LOOP
       BMI DEBUT
```

Ce programme boucle en émettant un bip continu sur le haut-parleur... pas passionnant n'est-ce-pas !

Maintenant tapez :

```
BRUN SHARE
BLOAD TEST1,A$300
SHARE,A$300,L50
```

...et votre oreille sera désormais chatouillée — en permanence — par un son strident et haché pendant la suite de votre travail. Vous pouvez lancer un programme Applesoft par exemple, cela ne l'arrêtera pas ! Le seul moyen :

*NOSHARE*

Ouf !

## Programmation

SHARE utilise les interruptions VBL de la carte souris qui doit être placée en slot 4. Il prend la main à chaque interruption, décompte alors le nombre d'impulsions déjà passées, et décide alors quel programme doit être commuté.

La commutation consiste à échanger la page 1 (pile du 6502) avec un buffer mémoire de même taille. On sauve aussi les différents registres et on sort de l'interruption par une instruction RTI. Si vous êtes sûr de vos programmes, vous pourrez envisager de n'échanger que la partie supérieure de la pile.

Attention SHARE ne sait gérer qu'un maximum de deux programmes simultanément. Par ailleurs aucun test d'erreur n'est effectué pour laisser le code le plus compact possible : donc pas d'incohérences sinon... plantage assuré.

## Optimisation

Lorsque deux programmes s'exécutent en même temps, il est évident qu'ils iront deux fois moins vite pour finir le même travail. Pas question donc de jouer aux Space Invaders !

Voici donc introduites les notions de :

- temps *elapsed* : temps total nécessaire ;
- temps *CPU* : temps pendant lequel le processeur fait effectivement tourner le programme ;
- temps *overhead* : temps nécessaire aux commutations et à la gestion du multi-tâches.

Les systèmes d'exploitation des très gros ordinateurs, qui gèrent simultanément plusieurs milliers de programmes, sont parfois considérés conune les systèmes les plus complexes créés par le cerveau humain. Vous n'aurez donc aucun mal à améliorer SHARE avec quelques fonctions supplémentaires...

### Fichiers sur la disquette Pom's

SHARE.CODE.S source Merlin-Pro en format text ;
SHARE assemblage du source, chaîné avec CMDLOAD par CMDLINK (voir Pom's 20) ;
TEST & TEST.S petit programme de test émettant un bip.

## Source TEST.S

### Assembleur Merlin-Pro

```
        org   $300
debut   ldx   £$80
loop    stx   $C030
        dex
        bpl   loop
        bmi   debut
```

## Récapitulation Test

*Ce programme n'est qu'un exemple de l'utilisation de SHARE. Il génère un 'bip' continu. Après activation de la routine par SHARE, A$300, L50 par exemple, cette routine sera exécutée par intermitence (d'où un bip haché) et l'ordinateur restera disponible pour toute autre application.*

*Après avoir saisi cette récapitulation sous moniteur, vous la sauvegarderez par : BSAVE TEST, A$300, 110*

```
0300:A2 80 8E 30 C0 CA 10 FA
0308:30 F6
```

## Source SHARE.CODE.S

### Assembleur Merlin-Pro

```
*******************************************
*                                         *
*          M U L T I - T A C H E          *
*                Avril 1987               *
*                                         *
*******************************************

*                        Commandes ProDOS :
*
*     ] SHARE,A$300,L33  33 % du temps alloué au
* ou  ] SHARE,A768,L33   programme en $300
*     ] NOSHARE          stoppe le mode multitâche
*     ] DOSHARE          annule l'effet de NOSHARE

        ORG  $2100       reloge par CMDLOAD

START   CLD              non nécessaire
        LDA  £>FIN+$100  fin
        LDA  £>LONG-$100
NON     LDA  $FF58       adresse de la commande ex
                         térieure précédente

COM     LDA  $BE6C       point d'entrée réel des c
                         ommandes
        STA  GETCHAR+1
        LDA  $BE6D
        STA  GETCHAR+2   initialise le pointeur ve
                         rs le buffer commande
        LDX  £1
        JSR  GETCHAR     charge le 1er caractère
        CMP  £'S'        et le compare...SHARE ?
        BEQ  S
        CMP  £'N'        ou bien nOSHARE ?
        BEQ  N1
        CMP  £'D'        ou dOSHARE ?
        BEQ  D1

ETNON   CLI              pas pour nous
        SEC
        JMP  (NON+1)     retour à la commande préc
                         édente...
D1      JMP  D
N1      JMP  N

GETCHAR LDA  $FFFF,X     charge un caractère du bu
                         ffer
        RTS

S       LDX  £1
        JSR  TEST        vérifie 'SHARE'
        BCS  ETNON

        LDA  £0
        STA  $BE53       commande externe
        LDA  £%00010000
        STA  $BE54       nom de fichier possible
        LDA  £%10010000
        STA  $BE55       parametre A et L
        LDA  £4          longueur de 'SHARE' - 1
        STA  $BE52
RAP     LDA  RAPPEL      indique à ProDOS où rappe
                         ler le programme
        LDA  RAP+1       après avoir extrait les p
                         aramètres
        STA  $BE50       (Adresse : A - Temps de p
                         artagé : L)
        LDA  RAP+2
        STA  $BE51
BEN     CLI
        CLC
        RTS              demande à ProDOS de trouv
                         er les paramètres

RAPPEL  LDA  $BE57       ProDOS revient par ici
        CMP  £$90        doit contenir 90
        BNE  BEN         si on a tout : adresse et
                          temps

        LDA  $BE58
        STA  ADB+1
        LDA  $BE59
        STA  ADH+1       A est transcrit vers le p
                         rogramme

        LDA  $BE60
        BNE  BEN         doit être 00
        LDA  $BE5F       charge le paramètre L (te
                         mps)

        CMP  £$21
        BEQ  Z21A0
        CMP  £$19
        BEQ  Z2193
        CMP  £$32
        BEQ  Z21AD
        CMP  £$4B
        BEQ  Z21C5
        CMP  £$42
        BEQ  Z21B8
        JMP  BEN

Z2193   LDA  £$03
        STA  TEMP0
        LDA  £$01
        STA  TEMP1
        JMP  BOK

Z21A0   LDA  £$02
        STA  TEMP0
        LDA  £$01
        STA  TEMP1
        JMP  BOK
```

SHARE FONCTIONNE SOUS PRODOS SUR LES Apple //e. IL FONCTIONNE AUSSI SUR LES Apple //c QUI ONT LA SOURIS EN PORT 4.

```
Z21AD    LDA  £$01
         STA  TEMP0
         STA  TEMP1
         JMP  BOK

Z21B8    LDA  £$01
         STA  TEMP0
         LDA  £$02
         STA  TEMP1
         JMP  BOK

Z21C5    LDA  £$01
         STA  TEMP0
         LDA  £$03
         STA  TEMP1

BOK      JSR  BEGIN       enfin lance le multitâche
         CLI              autorise interruptions
         JMP  RETOUR      fin

N        INX              teste le 'NOSHARE'
         JSR  GETCHAR
         CMP  £'O'
         BNE  ETNON1
         INX
         JSR  TEST
         BCS  ETNON1

NOSHARE  LDA  $C412
         STA  M_NO0+1
         LDX  £$C4
         LDY  £$40
         LDA  £$01        empêche interruptions VBL
M_NO0    JSR  $C400       va en SETMOUSE

         JMP  RETOUR

D        INX              teste le 'DOSHARE'
         JSR  GETCHAR
         CMP  £'O'
         BNE  ETNON1
         INX
         JSR  TEST        teste presence de SHARE
         BCS  ETNON1

DOSHARE  LDA  $C412
         STA  M_DO0+1
         LDX  £$C4
         LDY  £$40
         LDA  £$09        restaure interrupt. VBL
M_DO0    JSR  $C400       va en SETMOUSE

         JMP  RETOUR
ETNON1   JMP  ETNON

TEST     LDY  £0
TEST1    JSR  GETCHAR
         CMP  TAMP,Y
         BNE  ERR
         INX
         INY
         CPY  £5
         BNE  TEST1
OK       CLC
         RTS
ERR      SEC
         RTS

RETOUR   LDA  £0
         STA  $BE54
         STA  $BE55
         STA  $BE53
AD2      LDA  OK
         LDA  AD2+1
         STA  $BE50
         LDA  AD2+2
         STA  $BE51
         CLC
         RTS

BEGIN    LDA  $C419       INITMOUSE
         STA  MOUSE1+1

         LDA  $C412       SETMOUSE
         STA  MOUSE2+1

         LDA  $C413       SERVEMOUSE
         STA  MOUSE3+1

ADR      LDA  DEBUT       permet relogt (ProDos)
         LDA  ADR+2       poids fort
         LDX  ADR+1       poids faible

         STA  $3FF        initialise le vecteur des
                           interruptions
         STX  $3FE        vers la routine.

ADH      LDA  £$60        initialise la pile de tel
                          le manière que RTI
         PHA              provoque un saut en $6000
                          (2ème programme)
ADB      LDA  £$00        Ceci pour le démarage.
         PHA              La pile doit être du type
                           <registre d'etat>
         PHP              et <compteur ordinal>

         LDA  TEMP0       initialise le partagé du
                          temps machine pour
         STA  OUI0        la boucle...
         LDA  TEMP1
         STA  OUI1

         TSX              sauve le pointeur de pile
                          toujours pour
         STX  SS0         le démarage du programme
         STX  SS1

         LDX  £0

LOOPBIS  LDA  $100,X      boucle pour copier la pil
                          e dans les
         STA  PZ0,X       buffers du programme
         STA  PZ1,X
         INX
         BNE  LOOPBIS

         PLA              annule l'effet des empile
                          ments précédents
         PLA              L'ancienne pile a été sau
                          vegardée avant
         PLA

         LDA  £0          le 1er programme va s'exé
                          cuter
         STA  NUM

         LDX  £$C4
         LDY  £$40
MOUSE1   JSR  $C4BC       appele les routines souri
                          s (INITMOUSE)

         LDX  £$C4
         LDY  £$40
         LDA  £$09
MOUSE2   JSR  $C4B3       appele SETMOUSE demande l
                          es interruptions VBL

         CLI
         RTS              revient comme s'il ne s'é
                          tait rien passé

DEBUT    SEI              adresse d'arrivée d'une i
                          nterruption
         STX  SAVX        sauve les registres (Acc
```

```
                                  est en $45)
         STY   SAVY

         LDX   £$C4
         LDY   £$40
MOUSE3   JSR   $C4C4             appele SERVEMOUSE

         LDX   SAVX              et restaure les registres
         LDY   SAVY


DEBUT2   LDA   NUM               quel programme s'exécutai
                                  t ?
         BEQ   UT0               c'est le 1er, va en UT0

UT1      DEC   OUI1              c'est le second.Faut il c
                                  hanger ?
         BNE   RET               si non, reviens
         LDA   TEMP1             il faut changer, remet le
                                   pointeur de
         STA   OUI1              boucle a sa valeur maxima
                                  le

         LDA   $45               restaure l'accumulateur
         STA   SA1               et sauve tous les registr
                                  es pour le programme 1
         STX   SX1
         STY   SY1
         TSX
         STX   SS1               ainsi que le pointeur de
                                  pile

         LDX   £$00

LOOP0    LDA   $100,X            boucle transférant la pil
                                  e vers son buffer (1)
         STA   PZ1,X

         LDA   PZ0,X             et transfert le buffer (0
                                  ) vers la pile
         STA   $100,X
         INX
         BNE   LOOP0

         LDA   £0                et indique que le 1er pro
                                  gramme va s'exécuter
         STA   NUM

         LDA   SA0               restaure les registres du
                                   1er programme
         LDX   SS0
         TXS                     sans oublier la pile
         LDX   SX0
         LDY   SY0
         RTI                     retour d'interruption (ex
                                  écute le programme 1)

RET      LDA   $45               arrive ici s'il faut lais
                                  ser s'exécuter le
         RTI                     programme plus longtemps

UT0      DEC   OUI0              le programme 1 vient de s
                                  'exécuter
         BNE   RET               faut il encore le laisser
                                   faire ? Oui, en RET
         LDA   TEMP0
         STA   OUI0
         LDA   $45               on vient d'exécuter le 0
         STA   SA0
         STX   SX0
         STY   SY0               on sauve les registres
         TSX
         STX   SS0

         LDX   £$00

LOOP4    LDA   $100,X            transfert la pile vers le
                                   buffer
```


Nouveaux tarifs

# Speedisk™, quand il faut choisir...

À l'heure où votre Apple // est un peu à l'étroit dans ses disquettes, un choix objectif s'impose. Quelques cases vides vous permettront de mesurer l'apport de **Speedisk™**.

**Speedisk™** est un lecteur de disque – virtuel – qui conserve vos fichiers grâce à sa batterie et vous les restitue quasi-instantanément.
**Speedisk™** est une carte électronique sans pièce en mouvement, vue par votre ordinateur comme tout autre volume ProDOS, mais ultra-rapide et fiable.

| À vous de compléter : | Speedisk | ... |
|---|---|---|
| Accès à l'information | 0,2 ms | |
| Blocs ProDOS lus en 1 s | 100 | |
| Booter sur Basic.System | ≈3 s | |
| BRUN FILER | ≈ 0,5 s | |
| BRUN APPLEWRITER | ≈ 0,5 s | |
| Sauver un texte de 46Ko depuis AppleWriter | ≈ 5 s | |
| Lancer AppleWorks | ≈1 s | |
| Écrire un enregistrement 'Text' de 100 octets | ≈ 0,03 s | |
| Booter sur le Finder | ≈ 15 s | |
| Passer de Launcher à GSPaint | ≈17 s | |
| 'trajet' ProDOS 16, Launcher, MouseDesk, GSWrite | ≈ 34 s | |
| Capacité mémoire | 1 024Ko | |
| Tarif TTC | 4 490,00 F | |

**Speedisk™** existe également en version 384Ko et peut-être dotée d'une option horloge compatible ProDOS.
**Speedisk™** fonctionne sur les Apple ][+, Apple //e et Apple IIGS, elle occupe un seul connecteur et sa batterie (charge automatique) garde les données deux mois ordinateur hors tension. La carte est garantie un an.

Plus de renseignements ? Vous souhaitez une documentation ? une aide pour la comparaison ?
Utilisez ce bon :

Nom :

.....

Adresse :

.....

.....

LOGMA – 12, rue d'Anjou – 78000 Versailles (1) 39 51 24 43

```
        STA  PZ0,X

        LDA  PZ1,X        et le buffer vers la pile
        STA  $100,X
        INX
        BNE  LOOP4

        LDA  £1
        STA  NUM

        LDA  SA1
        LDX  S51          restaure tous les pointeu
                          rs
        TXS
        LDX  SX1
        LDY  SY1
        RTI               exécute le 1....

        BRK               pour le relogeur (partie
                          données)

TAMP    ASC  'SHARE'      pour l'interpréteur

NUM     DS   1            qui vient d'être exécuté
                          ?

SAVX    DS   1
SAVY    DS   1

PZ0     DS   256          sauvegarde la page 0 et p
                          ile du 1
SA0     DS   1
SX0     DS   1
SY0     DS   1
SS0     DS   1

PZ1     DS   256          sauvegarde la page 0 et p
                          ile du 2
SA1     DS   1
SX1     DS   1
SY1     DS   1
SS1     DS   1

OUI0    DS   1            pointeur interne (cycles)
OUI1    DS   1

TEMP0   DS   1            nombre de cycles pour le
                          prg
TEMP1   DS   1            nombre de cycles pour le
                          basic

FIN     =    *
LONG    =    FIN-START+$100
```

# Récapitulation Share

*Après avoir saisi cette récapitulation sous moniteur, vous la sauvegarderez par :*
*BSAVE SHARE, A$2000, L1536*

```
2000:AD 00 BF C9 4C F0 05 A9
2008:87 4C ED FD AD 4D BE F0
2010:05 A9 15 4C 09 BE AD 04
2018:21 69 00 20 98 20 90 05
2020:A9 DE 4C 09 BE CD 02 21
2028:90 F6 AE 08 BE 8D 08 BE
2030:8E 07 21 AE 07 BE 8E 06
2038:21 A0 00 8C 07 BE 48 E9
2040:21 85 3C 68 38 E9 04 85
2048:74 A9 21 85 49 84 48 A0
2050:00 B1 48 F0 27 20 8E F8
2058:A4 2F C0 02 D0 0F B1 48
2060:C9 21 90 09 CD 02 21 B0
2068:04 65 3C 91 48 A5 48 38
2070:65 2F 85 48 A5 49 69 00
2078:85 49 D0 D3 A0 00 A9 21
2080:84 3C 85 3D 18 6D 04 21
2088:84 42 88 84 3E 85 3F AD
2090:08 BE 85 43 C8 4C 2C FE
2098:8D FB 20 A5 74 18 69 04
20A0:8D FC 20 86 3D CE FC 20
20A8:F0 47 AD FC 20 8D FD 20
20B0:AD FD 20 48 4A 4A 4A AA
20B8:68 29 07 A8 B9 F3 20 3D
20C0:58 BF D0 E1 A5 3D D0 09
20C8:B9 F3 20 1D 58 BF 9D 58
20D0:BF AD FC 20 38 CE FD 20
20D8:ED FD 20 CD FB 20 D0 D0
20E0:A5 3D D0 07 18 AE FD 20
20E8:E8 8A 60 A9 00 85 3D F0
20F0:B9 38 60 80 40 20 10 08
20F8:04 02 01 00 00 00 AD 99
2100:D8 A9 26 A9 04 AD 58 FF
2108:A0 6C BE 8D 31 21 AD 6D
2110:BE 8D 32 21 A2 01 20 30
2118:21 C9 53 F0 17 C9 4E F0
2120:0C C9 44 F0 05 58 38 6C
2128:06 21 4C F6 21 4C D6 21
2130:BD FF FF 60 A2 01 20 19
2138:22 B0 EA A9 00 8D 53 BE
2140:A9 10 8D 54 BE A9 90 8D
2148:55 BE A9 04 8D 52 BE AD
2150:61 21 AD 50 21 8D 50 BE
2158:AD 51 21 8D 51 BE 58 18
2160:60 AD 57 BE C9 90 D0 F6
2168:AD 58 BE 8D 6E 22 AD 59
2170:BE 8D 6B 22 AD 60 BE D0
2178:E5 AD 5F BE C9 21 F0 20
2180:C9 19 F0 0F C9 32 F0 25
2188:C9 4B F0 39 C9 42 F0 28
2190:4C 5E 21 A9 03 8D 57 25
2198:A9 01 8D 58 25 4C CF 21
21A0:A9 02 8D 57 25 A9 01 8D
21A8:58 25 4C CF 21 A9 01 8D
21B0:57 25 8D 58 25 4C CF 21
21B8:A9 01 8D 57 25 A9 02 8D
21C0:58 25 4C CF 21 A9 01 8D
21C8:57 25 A9 03 8D 58 25 20
21D0:49 22 58 4C 2D 22 E8 20
21D8:30 21 C9 4F D0 38 E8 20
21E0:19 22 B0 32 AD 12 C4 8D
21E8:F1 21 A2 C4 A0 40 A9 01
21F0:20 00 C4 4C 20 22 E8 20
21F8:30 21 C9 4F D0 18 E8 20
2200:19 22 B0 12 AD 12 C4 8D
2208:11 22 A2 C4 A0 40 A9 09
2210:20 00 C4 4C 2D 22 4C 25
2218:21 A0 00 20 30 21 D9 45
2220:23 D0 08 E8 C8 C0 05 D0
2228:F2 18 60 38 60 A9 00 8D
2230:54 BE 8D 55 BE 80 53 BE
2238:AD 29 22 AD 39 22 8D 50
2240:BE AD 3A 22 8D 51 BE 18
2248:60 AD 19 C4 8D 9F 22 AD
2250:12 C4 8D A8 22 AD 13 C4
2258:8D B8 22 AD AC 22 AD 5D
2260:22 AE 5C 22 8D FF 03 8E
2268:FE 03 A9 60 48 A9 00 48
2270:08 AD 57 25 8D 55 25 AD
2278:58 25 8D 56 25 BA 8E 50
2280:24 8E 54 25 A2 00 BD 00
2288:01 9D 4D 23 9D 51 24 E8
2290:D0 F4 68 68 68 A9 00 8D
2298:4A 23 A2 C4 A0 40 20 BC
22A0:C4 A2 C4 A0 40 A9 09 20
22A8:B3 C4 58 60 78 8E 4B 23
22B0:8C 4C 23 A2 C4 A0 40 20
22B8:C4 C4 AE 4B 23 AC 4C 23
22C0:AD 4A 23 F0 41 CE 56 25
22C8:D0 39 AD 58 25 8D 56 25
22D0:A5 45 8D 51 25 8E 52 25
22D8:8C 53 25 BA 8E 54 25 A2
22E0:00 BD 00 01 9D 51 24 BD
22E8:4D 23 9D 00 01 E8 D0 F1
22F0:A9 00 8D 4A 23 AD 4D 24
22F8:AE 50 24 9A AE 4E 24 AC
2300:4F 24 40 A5 45 40 CE 55
2308:25 D0 F8 AD 57 25 8D 55
2310:25 A5 45 8D 4D 24 8E 4E
2318:24 8C 4F 24 BA 8E 50 24
2320:A2 00 BD 00 01 9D 4D 23
2328:BD 51 24 9D 00 01 E8 D0
2330:F1 A9 01 8D 4A 23 AD 51
2338:25 AE 54 25 9A AE 52 25
2340:AC 53 25 40 00 53 48 41
2348:52 45 00 00 00 00 00 00
2350:00 00 00 00 00 00 00 00
2358:00 00 00 00 00 00 00 00
2360:00 00 00 00 00 00 00 00
2368:00 00 00 00 00 00 00 00
2370:00 00 00 00 00 00 00 00
2378:00 00 00 00 00 00 00 00
2380:00 00 00 00 00 00 00 00
2388:00 00 00 00 00 00 00 00
2390:00 00 00 00 00 00 00 00
2398:00 00 00 00 00 00 00 00
23A0:00 00 00 00 00 00 00 00
23A8:00 00 00 00 00 00 00 00
23B0:00 00 00 00 00 00 00 00
23B8:00 00 00 00 00 00 00 00
23C0:00 00 00 00 00 00 00 00
23C8:00 00 00 00 00 00 00 00
23D0:00 00 00 00 00 00 00 00
23D8:00 00 00 00 00 00 00 00
23E0:00 00 00 00 00 00 00 00
23E8:00 00 00 00 00 00 00 00
23F0:00 00 00 00 00 00 00 00
23F8:00 00 00 00 00 00 00 00
2400:00 00 00 00 00 00 00 00
2408:00 00 00 00 00 00 00 00
2410:00 00 00 00 00 00 00 00
2418:00 00 00 00 00 00 00 00
2420:00 00 00 00 00 00 00 00
2428:00 00 00 00 00 00 00 00
2430:00 00 00 00 00 00 00 00
2438:00 00 00 00 00 00 00 00
2440:00 00 00 00 00 00 00 00
2448:00 00 00 00 00 00 00 00
2450:00 00 00 00 00 00 00 00
2458:00 00 00 00 00 00 00 00
2460:00 00 00 00 00 00 00 00
2468:00 00 00 00 00 00 00 00
2470:00 00 00 00 00 00 00 00
2478:00 00 00 00 00 00 00 00
2480:00 00 00 00 00 00 00 00
2488:00 00 00 00 00 00 00 00
2490:00 00 00 00 00 00 00 00
2498:00 00 00 00 00 00 00 00
24A0:00 00 00 00 00 00 00 00
24A8:00 00 00 00 00 00 00 00
24B0:00 00 00 00 00 00 00 00
24B8:00 00 00 00 00 00 00 00
24C0:00 00 00 00 00 00 00 00
24C8:00 00 00 00 00 00 00 00
24D0:00 00 00 00 00 00 00 00
24D8:00 00 00 00 00 00 00 00
24E0:00 00 00 00 00 00 00 00
24E8:00 00 00 00 00 00 00 00
24F0:00 00 00 00 00 00 00 00
24F8:00 00 00 00 00 00 00 00
2500:00 00 00 00 00 00 00 00
2508:00 00 00 00 00 00 00 00
2510:00 00 00 00 00 00 00 00
2518:00 00 00 00 00 00 00 00
2520:00 00 00 00 00 00 00 00
2528:00 00 00 00 00 00 00 00
2530:00 00 00 00 00 00 00 00
2538:00 00 00 00 00 00 00 00
2540:00 00 00 00 00 00 00 00
2548:00 00 00 00 00 00 00 00
2550:00 00 00 00 00 00 00 00
2558:00 00 85 DE A9 A3 8D 61
2560:9F 20 73 F2 20 00 9A 20
2568:F4 9F 20 17 9A 20 6C 9F
2570:4C BA 9A 20 62 9F 2C 46
2578:BE 10 03 4C 67 9D 2C 44
2580:BE 10 03 4C 16 9C 2C 43
2588:BE 30 24 A5 33 C9 DD F0
2590:C7 20 00 9A A0 3E B8 AC
2598:40 BE 20 10 FD C9 88 D0
25A0:0B A5 33 C9 BE F0 03 8A
25A8:F0 EA A9 88 4C 8D 9A 91
25B0:28 A5 33 C9 BE F0 3B AD
25B8:00 C0 10 11 C9 83 D0 0D
25C0:20 FB B2 AD 42 BE D0 39
25C8:8D 10 C0 F0 34 20 8A 9D
25D0:20 6C 9C B0 25 20 62 9F
25D8:90 04 CA BD 00 02 09 80
25E0:9D 00 02 8A D0 F4 20 00
25E8:9A 20 D5 9A 2C 53 BE 10
25F0:DC 60 20 8A 9D 20 48 9C
25F8:90 F7 20 45 B2 C9 05 D0
```

*Sylvie Gallet*

# Alarme

## un Classic Desk Accessory

**Au-delà de son caractère utilitaire, cet accessoire de bureau 'classique' pour Apple IIGS formera certainement un tremplin pour la réalisation de nombreux autres CDAs. À vos idées !**

Le programme proposé ici est un petit accessoire de bureau classique à deux titres : c'est un CDA et sa fonction est de vous doter d'un nouveau réveil-matin qui peut — si vous avez le sommeil (très) léger — vous tirer des bras de Morphée ou, plus certainement vous avertir qu'il est temps d'abandonner votre ordinateur favori pour vaquer à d'autres occupations (nécessairement moins passionnantes)...

La programmation de la sonnerie peut se faire sur les 24 heures qui suivent l'heure courante, l'arrêt de la sonnerie pouvant être manuel ou automatique.

## Installation

'Alarme' est un accessoire de bureau classique accessible comme le tableau de bord par Control-Ó-Esc à condition qu'il ait été placé dans le catalogue `SYSTEM/DESK.ACCS` d'une disquette contenant ProDOS 16.

Il est installé automatiquement dans le menu des CDAs en même temps que les autres CDAs lors du chargement du système et de l'application choisie.

## Utilisation Description

L'utilisation en est fort simple :

1 Appeler le menu des CDAs par Control-Ó-Esc.
2 Sélectionner 'Alarme programmable'.

L'écran qui apparaît est identique aux écrans du tableau de bord : dans un cadre (superbe), on trouve :

- le nom de l'accessoire ;
- les 4 réglages possibles ;
- la date et l'heure mises à jour en permanence (le format d'affichage des heures est celui choisi dans le 'Tableau de Bord/Horloge' : mode 24 heures ou mode AM/PM. En prime, vous avez droit au jour de la semaine) ;
- une ligne de commandes indiquant les touches utilisables.

La sélection d'une option se fait par les flèches ↑ et ↓.

La modification d'une option se fait par les flèches ← et →.

*Note : les modifications sont prises en compte immédiatement: la sonnerie peut donc se déclencher et s'arrêter lorsqu'on change les réglages des heures ou des minutes.*

Pour valider les réglages, appuyer sur ↵.

Pour annuler et revenir aux réglages précédents, sortir par Esc.

À la première utilisation, les réglages par défaut sont les suivants :


**SOFTWORLD**

17, Av. Emile Zola, 75015 Paris. TéL 40 59 02 99

| Langages | Prix TTC |
|---|---|
| Basic Compiler 1.0 Microsoft | 1650 F |
| Light Speed C Think | 1495 F |
| Light Speed Pascal Think | 1095 F |
| Turbo Pascal Borland | 750 F |
| Prolog Adv.A.I. Systems | 2250 F |
| ZBasic 4.01 Zedcor | 985 F |
| ZBasic Construction Set | 595 F |

CATALOGUE GRATUIT SUR DEMANDE

**Bon de Commande** à envoyer à SOFTWORLD

Nom : Prénom :
Adresse :
Code : Ville : Tél :
Qté : Logiciels : Prix TTC
Frais d'envoi :
Total TTC
Frais d'envoi recommandé : 30 Frs par 1000 Frs Mode de paiement : Chèque Par Carte bleue: téléphoner

THE PROFESSIONAL PROGRAMMER'S EXTENDER™

**Editeur : Invention Software**

Version dotée du code source à 100%

Tous les secrets de programmation sont révélés

Prix TTC 4500 Frs

```
Sonnerie :  arrêt
Arrêt :     manuel
Heures :    0
Minutes :   0
```

Par la suite, les réglages par défaut sont ceux qui ont été validés lors de la précédente utilisation.

# Sous le capot...

## ...les CDAs

Un accessoire de bureau classique est destiné à fonctionner dans un environnement... classique (non basé sur le concept d'événements et de 'bureau électronique') : contrairement aux NDAs accessibles par le menu ⌘ du bureau (qui sont activés soit quand leur fenêtre est au dessus du bureau, soit périodiquement selon une fréquence définie dans le bloc d'identification et sans intervention de l'utilisateur), un CDA prend le plein contrôle de la machine pendant une interruption provoquée par la frappe au clavier de Control–⌂–Esc.

Chaque fois que l'utilisateur appuie sur Control–⌂–Esc, et dès que le système est disponible, le Desk Manager sauve les pages TEXT des bancs 0, 1, $E0, et $E1 ainsi que les pages 0 et 1 du banc 0 (plus quelques babioles pour son usage personnel) et affiche le menu des CDAs qui contient les deux CDAs résidents (Tableau de Bord et Affichage Secondaire) et les CDAs (11 au maximum) chargés en RAM.

Le CDA choisi est activé et il dispose alors de toutes les ressources de la machine, avec l'obligation de gérer lui-même ses entrées/sorties. Il doit bien sûr veiller à ne pas perturber la mémoire utilisée par l'application interrompue, utiliser avec prudence des zones-mémoire qui ne lui appartiennent pas (mais, si cela est nécessaire, il peut demander au Desk Manager de lui attribuer de la mémoire supplémentaire).

Lorsqu'on quitte le CDA, le Desk Manager réaffiche le menu des CDAs et, si on quitte ce menu, il restaure les pages sauvées au début puis il rend la main à l'application.

## Structure d'un CDA

Un CDA comprend un bloc d'identification suivi du programme, le tout enregistré dans un fichier de type $B9 et placé dans le catalogue *SYSTEM/DESK.ACCS*.

Le bloc d'identification regroupe :

- le nom de l'accessoire : une chaîne de type pascal (c'est-à-dire précédée de sa longueur) : `str'Alarme programmable'`;
- un pointeur (4 octets) sur l'unique point d'entrée du programme : `dc i4'MAIN'`;
- un pointeur (4 octets) sur la routine de fermeture de l'accessoire : `dc i4'FIN'`. Cette routine est appelée chaque fois que ProDOS ou une application appelle `DeskShutDown`.

## Le programme

Il est appelé par le DeskManager en mode natif pur (registres et mémoire sur 16 bits). Il commence par sauvegarder le banc de données et le rendre égal au banc de programme. Viennent ensuite la réalisation de la page d'accueil (dans le même mode 40 ou 80 colonnes que celui du menu des CDAs) et l'affichage des réglages par défaut.

On met ensuite en place la routine qui affichera la date et l'heure système et qui utilise l'interruption d'une seconde.

Arrive enfin la boucle principale qui lit toutes les commandes entrées au clavier et qui aiguille en conséquence :

- soit sur le traitement des flèches haut ou bas : simple déplacement de la barre en vidéo inverse ;
- soit sur celui des flèches gauche ou droite : bascule marche/arrêt ou manuel/automatique ou bien augmentation/diminution du nombre d'heures ou minutes. Chaque utilisation des flèches provoque la mise à jour des paramètres pour la sonnerie ;
- soit vers la sortie par Escape ou par Retum.

## La routine de fermeture

Elle remet la sonnerie sur arrêt si celle-ci est... sur marche. Remarquons que pour quitter une application et recharger ProDOS 16 lorsque Alarme est en marche, il est indispensable de rebooter (avec Control–⌂–Reset ou l'option `REBOOT` du menu `QUITTER`). Ne pas utiliser `PRÉslot` à partir du basic, cela se solderait par un plantage... tous les 1/4 de seconde (en effet, la zone mémoire où se trouvait Alarme est soit effacée si la nouvelle disquette contient moins d'accessoires, soit écrasée par un autre accessoire, soit écrasée par le même accessoire qui est d'abord chargé à son adresse définitive et ensuite relogé). Pour passer d'une application à l'autre, la commande Bye ou le Finder ne poseront pas de problème avec cet accessoire actif.

## Les interruptions

Alarme utilise deux interruptions :

- l'interruption d'une seconde pendant l'accès au CDA. Le vecteur est mis en place à l'entrée et l'ancien vecteur est restauré à la sortie. Cette interruption n'est autorisée que pendant l'attente d'une touche : dans ce cas, aucun affichage n'est en cours et on est en mode vidéo normal, la date et l'heure ne risquent donc pas d'être affichées en inverse ;
- l'interruption de 1/4 de seconde quand la sonnerie est sur marche : l'ancien vecteur est restauré quand on la met sur arrêt.

Le traitement des interruptions est très simple, l'environnement a déjà été sauvé par le gestionnaire

d'interruption ; il faut savoir toutefois que les routines utilisateur sont appelées en mode émulation et qu'il faut terminer en mettant la retenue à 0 (`clc`) en mettant l'interruption à 0 (instruction variable suivant la source d'interruption) et finir par un simple `rtl`. L'environnement est alors restauré et le programme peut reprendre son cours.

### Utilisation des outils du IIGS

Alarme utilise des fonctions de deux outils : Miscellaneous Tools (outils divers) et Text Tools. En voici la liste avec un bref descriptif :

**Miscellaneous Tools**

ReadBParam : permet de lire un octet de la RAM préservée par batterie et contenant les paramètres du système modifiables grâce au Tableau de Bord.

ReadTimeHex : lecture de la date et de l'heure système, le résultat est donné sur 8 octets.

ReadAsciiTime : lecture de la date et de l'heure en ASCII, la chaîne obtenue occupe toujours 20 octets, elle utilise le format d'affichage choisi par l'utilisateur.

SetVector : affecte une adresse de vecteur d'interruption.

GetVector : renvoie l'adresse d'un vecteur d'interruption.

IntSource : autorise ou inhibe certaines sources d'interruption.

FWEntry : permet l'appel d'une routine système Apple // à partir d'un programme en mode natif avec passage de valeurs pour les registres A, X, Y.

**Text Tools**

WriteString: affiche une chaîne de type Pascal sur le périphérique de sortie en service.

WriteChar : affiche un caractère sur ledit périphérique.

## À propos du source...

Tous les appels aux outils sont écrits grâce aux macros correspondantes (elles sont facilement reconnaissables au '_' qui précède leur nom). Les autres macros proviennent du fichier `M16.UTILITY`, ce sont :

```
pushword £n pea n
pushlong £n pea nù-16 (1)
            pea n
str 'chaine' dc i1'lg chaine'
             dc c'chaine'
writestr adr pea adrù-16 (1)
             pea adr
             _WriteString
long         rep £$30
             longa on (2)
             longi on
short        sep £$30
             longa off
             longi off
longm        rep £$20
             longa on
shortm       sep £$20
             longa off
native       clc
             xce
             long
```

(1) adrù-16 : est un opérande sur 2 octets représentant le banc de adr (octet fort : 0, octet faible : banc). Le code obtenu n'est pas toujours identique à celui qui est affiché dans le listing d'assemblage.

(2) longa on/off longi on/off : longa informe l'assembleur que les mots manipulés par A sont sur 16 bits (on) ou sur 8 bits (off).
longi : même chose pour les registres index X et Y.

Ces deux directives ne génèrent aucun code et des instructions rep £$... ou sep £$... sont nécessaires pour faire passer le microprocesseur dans le mode choisi.

Les paramètres fournis aux fonctions des outils et ceux qui sont retournés transitent par la pile : on commence par réserver de la place pour les paramètres de sortie (signalés par S dans le source) on empile ensuite les paramètres d'entrée (E). Au retour de la fonction, il ne reste plus qu'à récupérer les paramètres de sortie qui se trouvent au sommet de la pile.

## Saisie du programme

Dans ces pages et sur la disquette Pom's, vous trouverez :

- le source Alarme.Src créé à l'aide d'Orca/M
- le fichier Macasmcopy qui permet de générer le fichier de macros, d'assembler, linker, transformer Alarme en fichier de type $B9 et installer ce dernier dans le directory `SYSTEM/DESK.ACCS` du drive 1 et dans celui du drive 2.
- Alarme, l'objet dont le type est type $B9.

Sans assembleur, vous pourrez utiliser Alarme de la façon suivante :

- Si vous avez la disquette Pom's, copier le fichier Alarme dans le dossier de votre disquette de travail, puis démarrer sur cette disquette : l'accessoire est installé automatiquement.
- Si vous ne l'avez pas, sous Moniteur, taper le code Alarme listé ci-après puis le sauvegarder par :

```
create alarme, T$B9
bsave alarme, a$2000, 1307
    2,t$B9
```

## *Fichier Exec MACASMCOPY*

```
macgen alarme.src alarme.m
    acros alarme.macros .d1
    /macros/m16.=
asml alarme.src
filetype alarme cda
copy alarme desk.accs/alar
    me
copy alarme .d1/system/des
    k.accs/alarme
```

## Fichier ALARME.SRC
### Assembleur Orca/M

```
*****************************************************************
*                                                               *
*          Accessoire de bureau: Alarme programmable            *
*                 (C) Sylvie GALLET pour POM'S                  *
*                        Janvier 1988                           *
*                                                               *
*****************************************************************

        absaddr on
        mload alarme.macros
        keep alarme

*****************************************************************
*                                                               *
*                  Données identifiant le CDA                   *
*                                                               *
*****************************************************************

CDAident START

        str 'Alarme programmable'      nom de l'accessoire
        dc i4'main'                    adresse d'ouverture
        dc i4'fin'                     adresse de fermeture
        END

*****************************************************************
*                                                               *
*        toutes les données utilisées par le programme          *
*                                                               *
*****************************************************************

DONNEES  DATA

****************************** textes ******************************

; codes de controle:
; ==================

; les codes utilisés sont:
;        7: bip                        24: désactive caract souris
;       12: form feed                  25: home, sans effacer l'écran
;       14: vidéo normal               27: active caractères souris
;       15: vidéo inverse

inverse  dc i1'2,15,24'
normal   dc i1'2,24,14'

; définitions des lignes pour le tracé du cadre de l'écran:
; =========================================================

; caractères souris utilisés:
;       H  ($48)  = ←
;       J  ($4A)  = ↓
;       K  ($4B)  = ↑_
;       L  ($4C)  =
;       M  ($4D)  = ⌋
;       U  ($55)  = →
;       Z  ($5A)  = |
;       _  ($5F)  = |

ligne1   dc i1'44,24,14',c' ',38c'_',c' ',i1'15,27'

ligne2   dc i1'44',c'Z',i1'24,14',c' Alarms programmable '
         dc i1'15',17c' ',i1'27',c'_'

ligne3   dc i1'39',c'Z',38c'L'

bord     str '_Z'

ligne4   dc i1'43,28,24,14',c'Choix:'
         dc i1'15,27,28,72,28,85,28,74,28,75,28,28,24,14'
         dc c'Annuler:Esc',i1'28,28',c'OK:',i1'28,15,27,77'

ligne5   dc i1'46',c'_',i1'24,14',c' ',i1'15,27',38c'L',i1'24,14'

; cadre pour la date et l'heure:
; ==============================

cadre1   dc i1'22',20c'_',i1'15,27'
cadre2   dc i1'22',c'Z',20i1'28',c'_'
cadre3   dc i1'22',20c'L',i1'14,24'

; items et options correspondantes, heure et date:
; ================================================

sonnerie str 'Sonnerie:'
son1     str ' arret '
son2     str ' marche'

arret    str 'Arret:'
arr1     str ' manuel     '
arr2     str ' automatique'

heure    str 'Heures:'
minute   str 'Minutes:'
numstr   str ' 00 AM'          pour afficher heures et minutes

; buffer pour _ReadAsciiTime

date     str ' 01/01/87'          ne pas séparer ces 2 instructions

time     str '12:00:00    '       la longueur de time est recouverte
         dc i1'25'                d'un espace ($A0) par _ReadAsciiTime

; jours de la semaine:
; ====================

di       str 'dimanche'
lu       str '  lundi '
ma       str '  mardi '
me       str 'mercredi'
je       str '  jeudi '
ve       str 'vendredi'
sa       str ' samedi '

****************** tables d'adresses et tabulations ******************

; tabulations:
; ============

t0       equ $0504
td0      equ $1009

; les tabulations sont de la forme: VTAB (msb) HTAB (lsb)

tabmsg   dc 12't0,t0+$100,t0+$300,t0+$400'     sonnerie arret heures
                                                minutes
         dc 12't0+9,t0+9,t0+$106,t0+$106'      marche arret manuel
                                                automatique
         dc 12't0+$307,t0+$408'                nombre heures nombre
                                                minutes
tabcadre dc 12'td0+$301,td0+$200,td0+$100,td0+1'
;                                              3 lign cadre date/heure
tabdate  dc 12'td0+$102'
tabtime  dc 12'td0+$205'                       date et heure système

; adresses:
; =========

adrmsg   dc 12'sonnerie,arret,heure,minute'    adresses des chaines
         dc 12'son1,son2,arr1,arr2'
adrcadre dc 12'cadre3,cadre2,cadre2,cadre1'
adrjour  dc 12'di,lu,ma,me,je,ve,sa'

******************* données et zones de stockage *******************

maxima   dc i2'24,60'        24 heures 60 minutes

item     ds 2

valeurs  ds 8                2 octets pour marche/arret
;                            2 autres pour manuel/automatique
;                            2 pour heures 2 pour minutes programmées

defauts  ds 8                valeurs non modifiées si sortie par esc
ampm     ds 2                indicateur du mode d'affichage de l'heure
;                              00 si mode AM PM, 01 si mode 24 h
temp     ds 2                adresse de stockage temporaire

         END

*****************************************************************
*                                                               *
*                          le programme                         *
*                                                               *
*****************************************************************

MAIN     START

************************ corps du programme ************************

wndlft   equ $0020           marge gauche et largeur écran (banc 0)

         using DONNEES

; initialisations:
; ================

; banc de données:
; ----------------
         phb                 sauve banc de données
         phk                 banc de programme := banc de données
         plb                 ces 3 instructions sont indispensables

; dimensions de la fenetre:
; -------------------------
         pushword £0         S
         pushword £$19       £ Paramètre n° $19 de la Battery Ram:
;                            contient le mode vidéo 40 ou 80 colonnes
         _ReadBParam         fonction $0C03 (Miscellaneous Tools)
         lda £$2800
         plx
         beq ec40col         si 0: 40 colonnes si 1: 80 colonnes
         ora £$0014          80 col -> marge gauche := 20, larg := 40
ec40col  sta >wndlft         40 col -> marge gauche := 0, larg := 40

; copie de defauts dans valeurs:
; ------------------------------
         ldx £defauts            adr source (16 bits poids faible)
         ldy £valeurs             "  destination  "    "    "
         lda £7                  nombre octets à déplacer (-1)
         mvn defauts,valeurs     bancs source, destination

; mode d'affichage des heures:
; ----------------------------
```

```
        pushword £0          S  place pour résultat
        pushword £$35        E  paramètre n° $35    la Battery Ram à
;                               lire : mode 24 h (01) ou AM PM (00)
        _ReadBParam          fonction $0C03 (Miscellaneous Tools)
        pla
        sta ampm

; réalisation de l'écran:
; =======================

notam   pushword £$C         efface l'écran
        _WriteChar
        lda £0
        jsr vhtab
        writestr ligne1      haut du cadre
        writestr ligne2
        writestr ligne3
        lda £$0002           A sert de compteur: on affiche la chaine
suivante pha                 bord composée de 2 caractères souris, de
        cmp £$0016           la ligne 2 à la ligne 21 comprise
        bcs basecran
        xba
        ora £$0027           VTAB := compteur, HTAB := 39 -> 1-er
        jsr vhtab            caractère sur le bord droit et 2-ème sur
        writestr bord        le bord gauche
        pla
        inc a
        bra suivante
basecran pla
        writestr ligne4      affiche la ligne de commandes
        lda £$1627
        jsr vhtab
        writestr ligne5      trait inférieur
        ldy £6               affiche cadre pour date et heure (4 lig)
e1      phy
        pea cadre10-16       E  pour _WriteString: adr haute
        lda adrcadre,y       E   "          "          "  basse
        pha
        lda tabcadre,y
        jsr vhtab
        _WriteString         fonction $1C0C (Text Tools)
        ply
        dey
        dey
        bpl e1
        lda £3               affiche les 4 options
        sta item
e2      jsr affitem
        dec item
        bpl e2
        lda valeurs          affiche 'marche' ou 'arret' suivant
        ora £4               valeurs -> A := 4 + valeurs
        jsr affopt           affiche la 5-iéme chaine (A = 4) ou la
;                            6-iéme (A = 5)
        lda valeurs+2        affiche 'manuel' ou 'automatique'
        ora £6               A := 6 + (0 ou 1) si A = 6: 'manuel'
        jsr affopt           si A = 7: 'automatique'
        ldx £4               valeurs,4 contient les heures programmées
        jsr affnum
        ldx £6               valeurs,6    "      "  minutes    "
        jsr affnum
        stz item             la barre inverse sera sur 'sonnerie:'

; changement vecteur interruption 1 seconde:
; ==========================================

        pushlong £0          S  place pour adresse ancien vecteur
        pushword £$15        E  n° de référence interruption 1 seconde
        _GetVector           fonction $1103 de Miscellaneous Tools
        pla                  on sauve cette adresse directemt dans la
        sta resls2+1         routine qui la restaurera
        pla
        sta resls1+1
        pushword £$15        E  encore lui...
        pushlong £afftime    E  adresse de notre routine
        _SetVector           fonction $1003 (Miscellaneous)
        jmp affinv           on peut enfin inverser 'sonnerie:'

; boucle principale: lecture du clavier:
; ======================================

boucle  lda £0
        jsr vhtab
        pushword £6          E  n° de réf. autoris. d'interrupt. 1 sec
        _IntSource           fonction $2303 (Miscellaneous)
        shortm
kbd     lda >$C000           lecture directe du clavier (cf article)
        bpl kbd
        xba                  cache le caractère lu le temps de...
        lda >$C010           ... remettre le clavier à 0 (bit ne
        xba                  supporte pas l'adressage long
        longm
        pha
        pushword £7          E  inhibition interruption 1 seconde
        _IntSource
        pla                  récupère le caractère lu
        and £$007F           masque les bits b15 à b7
        cmp £$1B             Esc ?
        bne t2
        bra sortiesc
t2      cmp £$0D             Return ?
        bne t3
        bra metajour
t3      cmp £$0A             Flèche bas ?
        bne t4
        brl descend
t4      cmp £$0B             "    haut ?
        bne t5
        brl monte
t5      cmp £$8              "    gauche ?
        bne t6
        brl augdim
t6      cmp £$15             "    droite ?
        bne emetbip
        brl augdim
emetbip pushword £7
        _WriteChar
        bra boucle

; sorties:
; =======

; 1 - par escape: toutes les modifications doivent etre annulées.
; ---------------

; le travail à effectuer dépend de l'état relatif de valeurs et de
; defauts:
; valeurs  defauts   somme   travail
; ---------------------------------------------
;  arret    arret      0     rien à faire
;  arret   marche      1     jsr onoff
; marche    arret      1     jsr onoff
; marche   marche      2     inhiber timer 1/4 s, mettre à jour les
;                            param de bsep, autoriser timer 1/4 s

sortiesc ldx £defauts+2      dans tous les cas, on reporte les 3
         ldy £valeurs+2      autres paramètres (mode d'arret, heures,
         lda £3              minutes) qui peuvent avoir été modifiés
         mvn defauts,valeurs
         lda valeurs
         clc
         adc defauts
         beq maj1            valeurs = defauts = 0
         cmp £1              somme = 1 ?
         bne escl            non <-> somme = 2
         jsr onoff           oui
         bra maj1
escl     pushword £5         si la sonnerie est sur marche, mieux vaut
         _IntSource          inhiber l'interruption le temps de
         jsr beepparm        modifier la routine qui la traite
         pushword £4
         _IntSource
         bra maj1

; 2 - par return:
; ---------------

metajour ldx £valeurs        les valeurs deviennent les defauts
         ldy £defauts
         lda £7
         mvn defauts,valeurs
maj1     pushword £$15       E  on replace l'ancien vecteur...
resls1   pea $0000           E  ...stocké ici  adr haute
resls2   pea $0000           E                 adr basse
         _SetVector
         plb                 restaure banc de données
         rtl

; routine de fermeture appelée en cas de DeskShutDown:
; ----------------------------------------------------

fin      ENTRY
         phb
         phk
         plb
         lda defauts         sonnerie sur 'arret' ?
         beq bye             oui -> rien à faire
         jsr beepoff         non -> on l'arrete
         stz defauts
bye      plb
         rtl

; traitement des 4 flèches:
; =========================

; déplacements de la barre en inverse:
; ------------------------------------

; affiche l'item sur lequel on se trouve en normal, descend ou monte
; le curseur selon le cas et affiche le nouvel item en inverse.

descend  jsr affitem
         inc item
         lda item
         cmp £$4             arrivé au 4-ème, on repart au 1-er
         bne affinv
         stz item
         bra affinv

monte    jsr affitem
         dec item
         bpl affinv          "    " 1-er "    "    " 4-ème
         lda £3
         sta item

affinv   writestr inverse
         jsr affitem
         writestr normal
```

**ALARME FONCTIONNE SOUS PRODOS 16 SUR LES Apple IIGS.**

```
        brl boucle

; changement de la valeur d'un item en fonction de la flèche (gauche
; ==================================
; ou droite) utilisée.

; en entrée: A = code ASCII de la flèche (08 ou $15)
; item est utilisé:
;       si 0: bascule marche/arret
;       si 1: bascule manuel/automatique
;       si 2: décrémente ou incrémente les heures
;       si 3:     "       "       "       "  minutes

augdim  tay                     Y = 8 ou $15
        lda item                A := 0..4
        bne ad0                 A <> 0
        jsr onoff               A = 0 (on est sur sonnerie)
        lda £00
        bra ad1
ad0     cmp £2                  A < 2 ?
        bcs heurmin             non A = 2,3
ad1     asl a                   oui -> A := A*2 (A := 0 ou A := 2)
        tax                     x := 0,2
        lda valeurs,x
        eor £1                  A := 1 - A
        sta valeurs,x           valeurs,x := 1 - valeurs,x
        txa
        ora valeurs,x           A := (0 ou 2) + (0 ou 1)
        ora £4                  A := A + 4 -> A = 4,5,6 ou 7
        jsr affopt              arret marche manuel automatique
        bra ad3
heurmin asl a                   -> A := 4 ou 6
        tax
        cpy £$15                flèche droite ?
        bcc diminue             non
augmente inc valeurs,x          oui
        lda valeurs,x
        cmp maxima-4,x          X = 4 ou 6 et il faut maxima ou maxima+2
;                               valeurs,X < maxima-4,X ?
        bcc ad2                 oui
        stz valeurs,x           non: 24 h -> 0 h et 60 mn -> 0 mn
        bra ad2
diminue dec valeurs,x
        bpl ad2
        lda maxima-4,x          valeurs,x < 0 -> on passe à 23 h ou 59 mn
        dec a
        sta valeurs,x
ad2     jsr affnum              affiche le nombre contenu dass valeurs,x
ad3     jsr beepparm            ajuste les paramètres de beep
        brl boucle

************************ sous-programmes ************************

; lecture date et heure en hexa:
; ==============================

; entrée: ries
; sortie: X := minutes système (lsb)
;         Y := heures     "     "
;         A := jour       "     "

readtima long
        pushlong £0             S  1-er mot: jour de la semaine 1..7
;                                          1 = dimanche    7 = samedi
;                               S  2-eme mot: mois jour
        pushlong £0             S  an heure minutes secondes
        _ReedTimaHex            fonction $0D03 (Miscellaneous)
        short                   1=m  1=x
        pla
        plx                     minutes
        ply                     heures
        pla
        pla
        pla
        pla                     nul
        pla                     jour de la semaine
        rts
        longa on                directives d'assemblage: ne générant pas
        longi on                de code

; mise en marche de la sonnerie si elle est sur arret (et inversement)
; ==============================

; avant l'appel de onoff, valeurs n'a pas été modifié:
;       si valeurs = 0  ->  mettre en marche
;       si valeurs = 1  ->  arreter

onoff   lda valeurs
        beq beepon
beepoff pushword £5             E  interruption 1/4 seconde inhibée pour
        _IntSource                 restaurer l'ancien vecteur...
        pushword £$E            E
rest4s  pushlong £0             E  ...modifié par beepon
        _SetVector
        rts

beepon  jsr beepparm
        pushword £5             E
        _IntSource
        pushlong £0             E  place pour vecteur
        pushword £$E            E  n° référence timar 1/4 seconde
        _GetVector
        pla
        sta rest4s+4
        pla
        sta rest4s+1
        pushword £$E            E
        pushlong £beep          E  notre adresse
        _SetVector
        pushword £4             E  timer 1/4 seconde autorisé
        _IntSource
        rts

; modifie des valeurs dans beep:
; ==============================
; heure de sonnerie, manuel/automatique, jourmeme

beepparm jsr readtime           au retour short (1=m 1=x)
        longa off
        longi off
        lda valeurs+2           0: manuel   1: automatique
        asl a
        asl a
        asl a
        asl a
        asl a
        asl a
        dec a                   A := $3F: (manuel) ou $FF: (automatique)
        sta auto+1
        lda valeurs+4           hlimite+1 := heures programmées
        sta hlimite+1
        lda valeurs+6           mlimite+1 := minutes
        sta mlimite+1
        tya                     y = heures système
        xba                     et x = minutes    "
        txa                     -> A := heures minutes système
        stz jourmeme+1
        long                    natif pur
        sta temp
        lda valeurs+4
        xba
        ora valeurs+6           -> A := heures minutes programmées
        cmp temp                heure programmée >= heure système ?
        bcs bprts               oui -> jourmeme+1 reste égal à 0
        inc jourmeme+1          non -> jourmeme+1 := 1 (jourmeme+2 n'est
bprts   rts                     pas affecté par cet inc)

; affichage de presque toutes les chaines:
; ==========================================

; en entrée:
;       pour affitem: le contenu de item (0..3)
;       pour affopt:  le contenu de A (4..7)
; qui permettent de calculer le numéro de la chaine à afficher.

affitem lda item
affopt  asl a                   A := 0..$E (par pas de 2)
        tay                     Y := 0..$E   "   "   "   "
        pea sonnerie-16         E  adresse chaine: poids fort
        lda adrmsg,y            lit dans la table adrmsg le poids faible
        pha                     E
        lda tabmsg,y
        jsr vhtab
        _WriteString            fonction $1C0C (Text Tools)
        rts

; affiche un nombre < à 60 sur 2 chiffres décimaux et précédé par
; ================================================================
; un espace. Si heure en format AM PM, affiche aussi AM ou PM.
;   Si le nombre est inférieur à 10, ajoute un espace en tête
; la chaine numstr est modifiée à chaque appel.
; en entrée: X = 4 ou 6 (4 pour les heures, 6 pour les minutes)

affnum  lda £'  '               deux espaces
        sta numstr+5            à l'emplacement de AM ou PM
        lda valeurs,x
        cpx £4                  affichage des heures ?
        bne mn                  non
        ldy ampm                oui -> es mode 24 heures ?
        bne mn                         oui
        ldy £'MA'                      non -> y := 'MA'
        cmp £$0C                A < 12 ?
        bcc matin               oui
        ldy £'MP'               non -> Y := 'MP'
        sbc £$0C                A := A - 12
matin   sty numstr+5            ->numstr+5 := 'A' ou 'P', numstr+6 := 'M'
mn      pha
        lda tabmsg+12,x
        jsr vhtab
        pla
        ldx £0                  X compteur de dizaines
afn1    cmp £$A                 A < 10 ?
        bcc afn2                oui
        sbc £$A                 non -> A := A - 10
        inx                        et X := X + 1
        bra afn1
afn2    xba                     A := unités 00
        sta numstr+2            temporaire
        txa                     A := 00 dizaines
        ora numstr+2            A := unités dizaines
        ora £$3030              -> A := codes ASCII des unités et dizain.
        cpx £0                  0 dizaine ?
        bne deuxch              non -> il y a deux chiffres
        eor £$0010              -> A := chiffre unités, espace
deuxch  sta numstr+2            lsb de A stocké avant msb -> diz en lar
        writestr numstr
        rts
```

```
; positionnement du curseur:
; ==========================

; en entrée: A = VTAB HTAB

vhtab   sta >$0024          A stocké dans $0024 (banc 00) à l'envers
;                           msb dans $25 (vtab) lsb dans $24 (htab)
        lda £$FC22          routine système VTAB
        pha                 S  place pour P (registre d'état)
        pha                 S    "    "   A
        pha                 S    "    "   X
        pha                 S    "    "   Y
        pha                 E  val A passée à $FC22, sans intéret ici
        pha                 E    " X   "       "     "     "     "
        pha                 E    " Y   "       "     "     "     "
        pha                 E  adresse de la routine appelée
        _FWEntry            fonction n° $2403 (Miscelleneous)
        pla                 récupération dans l'ordre de Y,X,A,P
        pla
        pla
        pla
        rts

************  routines de traitement des interruptions  *************

; (appelées en mode émulation, vitesse normale, banc de données = 00)

; interruption 1 seconde:
; =======================

afftime jsr readtime
        native
        asl a               A := 2..14
        tax                 X := 2..14
        pea diù-16          E  adr haute jour à afficher pour
                                    _WriteString
        lde >adrjour-2,x    E   "  basse   "  "     "
        pha
        lda >tabdate
        jsr vhtab
        _WriteString        affiche le jour
        pushlong £date+2    E  buffer où sera stockée la date
        _ReadAsciiTime      fonction $0F03 (Miscellaneous)
        writestr date
        lda >time           >time contient un espace ($A0) qu'il faut
        eor £$00AC          remplacer par $0C et >time+1 contient le
        sta >time           1er caractère de l'heure a conserver
        lda >tabtime
        tax
        lde >ampm           quel mode d'affichage des heures ?
        beq am              AM PM
        inx                 24 h -> on décale de 2 carectère à droite
        inx
am      txa
        jsr vhtab
        writestr time
        short
        lda £$40            pour autoriser à nouveau l'interruption
        trb $C032           on met à 0 le bit b6 de $C032 (SCANINT)
        clc                 interruption bien traitée
        rtl

; interruption 1/4 de seconde:
; ============================

; surveille l'heure et déclenche le beep.
;  En cas d'arret manuel, la sonnerie continue jusqu'à l'intervention
;  de l'utilisateur et l'interruption est désactivée.
;  En cas d'arret automatique, l'interruption n'est pas désactivée et
;  à l'accès suivant, le menu affiche 'marche'.

beep    jsr readtime
journeme lda £0             opérande modifié par beepparm:
;                                 0 si sonnerie le jour meme
;                                 1 si    "     le lendemain
        beq hlimite         si 0: on peut comparer les heures
        cpy £0              si 1: on surveille le passage de minuit
        bne pasbeep
        cpx £0
        bne pasbeep
        txa                 minuit passé -> journeme+1 := 0
        ste >journeme+1
hlimite cpy £0              opérande modifié par beepparm
        bcc pasbeep
mlimite cpx £0                 "        "     "     "
        bcc pasbeep
auto    lde £$FF            lui aussi: si $FF -> ne sera pas décrém.
;                           si $3F -> il sera décrémenté
        bmi loop0           à chaque passage
        beq pasbeep         arrivé à 0: arret de la sonnerie
        dec a
        sta >auto+1
loop0   lda £0              opérande modifié à chaque passage
        inc a               si le bit b1 de cet opérande est à 0: pas
        sta >loop0+1        de beep, s'il est à 1: beep -> 2 beeps 2
        lsr a               silences
        lsr a
        bcc pasbeep
        ldx £$40
loop1   ldy £$50
loop2   dey
        bne loop2
        bit $C030
        dex
        bne loop1
pasbeep sta $C047           l accès à $C047 (CLRVBLINT) autorise à
        clc                 nouveau l'interruption
        rtl

        END
```

# Fichier Alarme. Macros

```
 MACRO
&lab _SetVector
&lab ldx £$1003
 jsl $E10000
 MEND
 MACRO
&lab _GetVector
&lab ldx £$1103
 jsl $E10000
 MEND
 MACRO
&lab _IntSource
&lab ldx £$2303
 jsl $E10000
 MEND
 MACRO
&lab _FWEntry
&lab ldx £$2403
 jsl $E10000
 MEND
 MACRO
&lab _WriteString
&lab ldx £$1C0C
 jsl $E10000
 MEND
 MACRO
&lab str &string
&lab dc i1'1:&string',c'&s
        tring'
 MEND
 MACRO
&lab short
&lab ANOP
 sep £%00110000
 longa off
 longi off
 MEND
 MACRO
&lab writestr &addr
&lab ANOP
 AIF c:&addr=0,.c
 lclc &char
&char amid &addr,1,1
 AIF "&char"="£",.a
 pea &addrù-16
 pea &addr
 ago .b
.a
 lclc &str
&str amid &addr,3,1:&addr-
        3
 bra ¨b&syscnt
¨a&syscnt dc i1'1:&str',c'
        &str'
¨b&syscnt pea ¨a&syscntù-1
        6
 pea ¨a&syscnt
.b
 ldx £$1C0C
 jsl $E10000
 MEXIT
.c
 phy
 pha
 AGO .b
 MEND
 MACRO
&lab long
&lab ANOP
 rep £%00110000
 longa on
 longi on
 MEND
 MACRO
&lab _ReadAsciiTime
&lab ldx £$0F03
 jsl $E10000
 MEND
 MACRO
&lab  ReadTimeHex
&lab ldx £$0D03
 jsl $E10000
 MEND
 MACRO
&lab native &mode
&lab ANOP
 clc
 xce
 AIF c:&mode=0,.a
 &mode
 MEXIT
.a
 long
 MEND
 MACRO
&lab _ReadBParam
&lab ldx £$0C03
 jsl $E10000
 MEND
 MACRO
&lab pushword &sysopr
&lab ANOP
 AIF c:&sysopr=0,.b
 LCLC &C
 LCLC &REST
 LCLA &BL
&C AMID "&sysopr",1,1
&BL ASEARCH "&sysopr"," ",
        1
 AIF &BL>0,.a
&BL SETA L:&sysopr+1
.a
&REST AMID "&sysopr",2,&BL
        -2
 AIF ("&C"="£").AND.(S:LON
        GA),.immediate
 lda &sysopr
 pha
 MEXIT
.b
 pha
 MEXIT
.immediate
 dc I1'$F4',I2'&REST'
 MEND
 MACRO
&lab pushlong &addr,&offse
        t
&lab ANOP
 LCLC &C
 LCLC &REST
&C AMID &addr,1,1
 AIF "&C"="£",.immediate
 AIF "&C"="°",.zeropage
 AIF C:&offset=0,.nooffset
 AIF "&offset"="s",.stack
```

```
 pushword &addr+2,&offset
 pushword &addr,&offset
 MEXIT
.nooffset
 pushword &addr+2
 pushword &addr
 MEXIT
.immediate
&REST AMID &addr,2,L:&addr
        -1
 dc I1'$F4',I2'(&REST)ù-16
        *
 dc I1'$F4',I2'&REST'
 MEXIT
.stack
 pushword &addr+2,s
 pushword &addr+2,s
 MEXIT
.zeropage
 ldy £&offset+2
 pushword &addr,y
 ldy £&offset
 pushword &addr,y
 MEND
 MACRO
&lab _WriteChar
&lab ldx £$180C
 jsl $E10000
 MEND
 MACRO
&lab longm
&lab ANOP
 rep £%00100000
 longa on
 MEND
 MACRO
&lab shortm
&lab ANOP
 sep £%00100000
 longa off
 MEND
```

## Fichier alarme

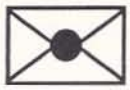
*Après avoir saisi cette récapitulation sous moniteur, vous la sauvegarderez par :*

```
CREATE ALARME,T$B9
BSAVE ALARME,A$2000,L3072,T$B9
```

```
2000:06 00 00 00 00 00 00 00 D9 06 00 00 00 0A 04 01
2010:00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
2020:00 00 01 00 00 00 00 00 2C 00 40 00 20 20 20 20
2030:20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
2040:F2 D9 06 00 00 13 41 6C 61 72 6D 65 20 70 72 6F
2050:67 72 61 6D 6D 61 62 6C 65 50 02 00 00 39 04 00
2060:00 02 0F 18 02 18 0E 2C 18 0E 20 5F 5F 5F 5F 5F
2070:5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F
2080:5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F
2090:5F 20 0F 1B 2C 5A 18 0E 20 41 6C 61 72 6D 65 20
20A0:70 72 6F 67 72 61 6D 6D 61 62 6C 65 20 0F 20 20
20B0:20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 1B
20C0:5F 27 5A 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
20D0:4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
20E0:4C 4C 4C 4C 4C 4C 4C 4C 02 5F 5A 2B 1C 18 0E
20F0:43 68 6F 69 78 3A 0F 1B 1C 48 1C 55 1C 4A 1C 4B
2100:1C 1C 18 0E 41 6E 6E 75 6C 65 72 3A 45 73 63 1C
2110:1C 4F 4B 3A 1C 0F 1B 4D 2E 5F 18 0E 20 0F 1B 4C
2120:4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
2130:4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
2140:4C 4C 4C 4C 4C 18 0E 16 5F 5F 5F 5F 5F 5F 5F 5F
2150:5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 0F 1B 16 5A
2160:1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C 1C
2170:1C 1C 1C 1C 5F 16 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
2180:4C 4C 4C 4C 4C 4C 4C 4C 4C 4C 0E 18 09 53 6F 6E
2190:6E 65 72 69 65 3A 07 20 61 72 72 65 74 20 07 20
21A0:6D 61 72 63 68 65 06 41 72 72 65 74 3A 0C 20 6D
21B0:61 6E 75 65 6C 20 20 20 20 20 0C 20 61 75 74 6F
21C0:6D 61 74 69 71 75 65 07 48 65 75 72 65 73 3A 08
21D0:4D 69 6E 75 74 65 73 3A 06 20 30 30 20 41 4D 09
21E0:20 30 31 2F 30 31 2F 38 37 0B 31 32 3A 30 30 3A
21F0:30 30 20 20 20 19 08 64 69 6D 61 6E 63 68 65 08
2200:20 20 6C 75 6E 64 69 20 08 20 20 6D 61 72 64 69
2210:20 08 6D 65 72 63 72 65 64 69 08 20 20 6A 65 75
2220:64 69 20 08 76 65 6E 64 72 65 64 69 08 20 73 61
2230:6D 65 64 69 20 04 05 04 06 04 08 04 09 0D 05 0D
2240:05 0A 06 0A 06 0B 08 0C 09 0A 13 09 12 09 11 0A
2250:10 0B 11 0E 12 47 01 61 01 82 01 8A 01 51 01 59
2260:01 68 01 75 01 30 01 19 01 19 01 02 01 B1 01 BA
2270:01 C3 01 CC 01 D5 01 DE 01 E7 01 18 00 3C 00 00
2280:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2290:00 00 00 00 00 8B 4B AB F4 00 00 F4 19 00 A2 03
22A0:0C 22 00 00 E1 A9 00 28 FA F0 03 09 14 00 8F 20
22B0:00 00 A2 44 02 A0 3C 02 A9 07 00 54 00 00 F4 00
22C0:00 F4 35 00 A2 03 0C 22 00 00 E1 68 8D 4C 02 F4
22D0:0C 00 A2 0C 18 22 00 00 E1 A9 00 00 20 0F 06 F4
22E0:00 00 F4 22 00 A2 0C 1C 22 00 00 E1 F4 00 00 F4
22F0:4F 00 A2 0C 1C 22 00 00 E1 F4 00 00 F4 7C 00 A2
2300:0C 1C 22 00 00 E1 A9 02 00 48 C9 16 00 B0 18 EB
2310:09 27 00 20 0F 06 F4 00 00 F4 A4 00 A2 0C 1C 22
2320:00 00 E1 68 1A 80 E2 68 F4 00 00 F4 A7 00 A2 0C
2330:1C 22 00 00 E1 A9 27 16 20 0F 06 F4 00 00 F4 D3
2340:00 A2 0C 1C 22 00 00 E1 A0 06 00 5A F4 00 00 B9
2350:20 02 48 B9 04 02 20 0F 06 A2 0C 1C 22 00 00 E1
2360:7A 88 88 10 E6 A9 03 00 8D 3A 02 20 97 05 CE 3A
2370:02 10 F8 AD 3C 02 09 04 00 20 9A 05 AD 3E 02 09
2380:06 00 20 9A 05 A2 04 00 20 B1 05 A2 06 00 20 B1
2390:05 9C 3A 02 F4 00 00 F4 00 00 F4 15 00 A2 03 11
23A0:22 00 00 E1 68 8D 2E 04 68 8D 2B 04 F4 15 00 F4
23B0:00 00 F4 2A 06 A2 03 10 22 00 00 E1 4C 6A 04 A9
23C0:00 00 20 0F 06 F4 06 00 A2 03 23 22 00 00 E1 E2
23D0:20 AF 00 C0 00 10 FA EB AF 10 C0 00 EB C2 20 48
23E0:F4 07 00 A2 03 23 22 00 00 E1 68 29 7F 00 C9 1B
23F0:00 D0 02 80 33 C9 0D 00 D0 02 80 64 C9 0A 00 D0
2400:03 82 8A 00 C9 0B 00 D0 03 82 95 00 C9 08 00 D0
2410:03 82 BB 00 C9 15 00 D0 03 82 B3 00 F4 07 00 A2
2420:0C 18 22 00 00 E1 80 97 A2 46 02 A0 3E 02 A9 05
2430:00 54 00 00 AD 3C 02 18 6D 44 02 F0 2F C9 01 00
2440:D0 05 20 00 05 80 25 F4 05 00 A2 03 23 22 00 00
2450:E1 20 60 05 F4 04 00 A2 03 23 22 00 00 E1 80 0C
2460:A2 3C 02 A0 44 02 A9 07 00 54 00 00 F4 15 00 F4
2470:00 00 F4 00 00 A2 03 10 22 00 00 E1 AB 6B 8B 4B
2480:AB AD 44 02 F0 06 20 05 05 9C 44 02 AB 6B 20 97
2490:05 EE 3A 02 AD 3A 02 C9 04 00 D0 13 9C 3A 02 80
24A0:0E 20 97 05 CE 3A 02 10 06 A9 03 00 8D 3A 02 F4
24B0:00 00 F4 1C 00 A2 0C 1C 22 00 00 E1 20 97 05 F4
24C0:00 00 F4 1F 00 A2 0C 1C 22 00 00 E1 82 F0 FE A8
24D0:AD 3A 02 D0 08 20 00 05 A9 00 00 80 05 C9 02 00
24E0:B0 17 0A AA BD 3C 02 49 01 00 9D 3C 02 8A 1D 3C
24F0:02 09 04 00 20 9A 05 80 26 0A AA C0 15 00 90 10
2500:FE 3C 02 BD 3C 02 DD 32 02 90 11 9E 3C 02 80 0C
2510:DE 3C 02 10 07 BD 32 02 3A 9D 3C 02 20 B1 05 20
2520:60 05 82 9A FE C2 30 F4 00 00 F4 00 00 F4 00 00
2530:F4 00 00 A2 03 0D 22 00 00 E1 E2 30 68 FA 7A 68
2540:68 68 68 68 60 AD 3C 02 F0 1B F4 05 00 A2 03 23
2550:22 00 00 E1 F4 0E 00 F4 00 00 F4 00 00 A2 03 10
2560:22 00 00 E1 60 20 60 05 F4 05 00 A2 03 23 22 00
2570:00 E1 F4 00 00 F4 00 00 F4 0E 00 A2 03 11 22 00
2580:00 E1 68 8D 16 05 68 8D 13 05 F4 0E 00 F4 00 00
2590:F4 95 06 A2 03 10 22 00 00 E1 F4 04 00 A2 03 23
25A0:22 00 00 E1 60 20 E0 04 AD 3E 02 0A 0A 0A 0A 0A
25B0:0A 3A 8D B2 06 AD 40 02 8D AA 06 AD 42 02 8D AE
25C0:06 98 EB 8A 9C 99 06 C2 30 8D 4E 02 AD 40 02 EB
25D0:0D 42 02 CD 4E 02 B0 03 EE 99 06 60 AD 3A 02 0A
25E0:A8 F4 00 00 B9 10 02 48 B9 F0 01 20 0F 06 A2 0C
25F0:1C 22 00 00 E1 60 A9 20 20 8D 98 01 BD 3C 02 E0
2600:04 00 D0 16 AC 4C 02 D0 11 A0 41 4D C9 0C 00 90
2610:06 A0 50 4D E9 0C 00 8C 98 01 48 BD FC 01 20 0F
2620:06 68 A2 00 00 C9 0A 00 90 06 E9 0A 00 E8 80 F5
2630:EB 8D 95 01 8A 0D 95 01 09 30 30 E0 00 00 D0 03
2640:49 10 00 8D 95 01 F4 00 00 F4 93 01 A2 0C 1C 22
2650:00 00 E1 60 8F 24 00 00 A9 22 FC 48 48 48 48 48
2660:48 48 48 A2 03 24 22 00 00 E1 68 68 68 68 60 20
2670:E0 04 18 FB C2 30 0A AA F4 00 00 BF 26 02 00 48
2680:AF 0C 02 00 20 0F 06 A2 0C 1C 22 00 00 E1 F4 00
2690:00 F4 9C 01 A2 03 0F 22 00 00 E1 F4 00 00 F4 9A
26A0:01 A2 0C 1C 22 00 00 E1 AF A4 01 00 49 AC 00 8F
26B0:A4 01 00 AF 0E 02 00 AA AF 4C 02 00 F0 02 E8 E8
26C0:8A 20 0F 06 F4 00 00 F4 A4 01 A2 0C 1C 22 00 00
26D0:E1 E2 30 A9 40 1C 32 C0 18 6B 20 E0 04 A9 00 F0
26E0:0D C0 00 D0 34 E0 00 D0 30 8A 8F 99 06 00 C0 00
26F0:90 27 E0 00 90 23 A9 FF 30 07 F0 1D 3A 8F B2 06
2700:00 A9 00 1A 8F BD 06 00 4A 4A 90 0D A2 40 A0 50
2710:88 D0 FD 2C 30 C0 CA D0 F5 8D 47 C0 18 6B F5 04
2720:00 14 00 50 02 F5 04 00 18 00 39 04 F5 02 00 10
2730:02 47 01 F5 02 00 12 02 61 01 F5 02 00 14 02 82
2740:01 F5 02 00 16 02 8A 01 F5 02 00 18 02 51 01 F5
2750:02 00 1A 02 59 01 F5 02 00 1C 02 68 01 F5 02 00
2760:1E 02 75 01 F5 02 00 20 02 30 01 F5 02 00 22 02
2770:19 01 F5 02 00 24 02 19 01 F5 02 00 26 02 02 01
2780:F5 02 00 28 02 B1 01 F5 02 00 2A 02 BA 01 F5 02
2790:00 2C 02 C3 01 F5 02 00 2E 02 CC 01 F5 02 00 30
27A0:02 D5 01 F5 02 00 32 02 DE 01 F5 02 00 34 02 E7
27B0:01 F5 02 00 6E 02 44 02 F5 02 00 71 02 3C 02 F5
27C0:01 F0 77 02 3C 02 F5 01 F0 78 02 44 02 F5 02 00
27D0:88 02 4C 02 F5 02 00 98 02 0F 06 F5 02 F0 9B 02
```

*Suite page 24* ➡

# Plus avec AppleWorks :



FONCTIONNE SOUS PRODOS SUR LES Apple //e, //c et IIGS.

# Courrier automatique

Dimitri Geystor

**Vous êtes passé d'AppleWriter à AppleWorks et vous regrettez votre WPL ? Si vous avez adopté SuperMacroWorks, vous disposez non seulement de la souris mais de commandes automatiques programmables.**
**Les macro-commandes qui suivent, œuvres d'un spécialiste, automatisent votre courrier. Pour les //e, //c et IIGS.**

Chose promise, chose due, voici la macro-commande d'adressage automatique du courrier annoncée dans le numéro précédent[1].

## Courrier automatisé

Partant d'une Base de données qui contient les adresses de vos correspondants, cette macro exécute, en une seule commande, la mise en page d'une lettre :

- elle crée un fichier Traitement de texte ;
- elle place votre nom et votre adresse en haut à gauche ;
- elle inscrit l'adresse de votre correspondant à l'endroit idoine pour une enveloppe à fenêtre ;
- elle ajoute au-dessous la date ;
- elle place enfin le curseur au début du texte à taper.

Et, si vous avez l'œil rapide, vous remarquerez qu'elle en profite, mine de rien, pour donner au fichier de traitement de texte ainsi créé le nom de votre correspondant. De sorte que vous pouvez à tout moment sauver votre lettre sans risquer d'écraser la précédente.

Comme la place sur le Bureau est limitée à 12 fichiers, il est commode en outre de disposer d'une macro de 'nettoyage' qui efface tous les fichiers du Bureau, en vous donnant néanmoins la possibilité d'intervenir en cours de macro pour sauver d'abord ceux des fichiers que vous n'avez pas sauvegardés pendant votre travail. C'est la dernière des macros du fichier ADRESSAGE, ⌘-S.

## Le fonctionnement

Cet ensemble de macros utilise plusieurs mémoires. Nous allons les passer en revue, et voir comment elles sont mises à profit.

Tout d'abord — à tout seigneur tout honneur — la fameuse *Macro 0* ; vous la rencontrerez souvent, car c'est vraiment la mémoire à tout faire de SuperMacroWorks : le chargement et le déchargement des autres mémoires transite par elle, et de plus elle a quelques emplois bien particuliers. Prenons, par exemple, la première macro de ADRESSAGE, définie par la lettre C : c'est elle qui permettra de retrouver le fichier ADRESSES. DEMO à partir de n'importe quel endroit d'Appleworks.

Le principe est le suivant : la commande <menu> , ou ⇧-( au clavier, fait deux choses à la fois :

- elle enregistre l'identité du fichier en cours (c'est-à-dire celui qui est à l'écran) ;
- elle inscrit dans la mémoire de la Macro 0 l'ordre d'aller au Menu principal (si vous faites ⌘-0 à ce stade, vous vous retrouverez en effet dans ce Menu).

De son côté, la commande <resume>, ou ⇧-) au clavier, inscrit dans la Macro 0 l'ordre de revenir au fichier à partir duquel a été actionnée la commande <menu>. Faites ⇧-) suivi de ⌘-0, et vous allez vous retrouver au point de départ. Bien. Et si l'on fait <menu><resume> à la file ? On emmagasine simplement dans la Macro 0 l'ordre de revenir au fichier actuel.

Seulement la Macro 0, nous allons en avoir besoin pour autre chose entretemps ; alors, rangeons cet ordre que contient la Macro 0 dans une mémoire auxiliaire, la mémoire n° 9 par exemple. Cela s'écrit <save0>9. Comme ce remue-ménage de mémoires ne se voit absolument pas à l'écran, ajoutons un petit message pour avertir, et la macro ⌘-C sera complète...

Cette opération est à la base de tout rappel contrôlé des fichiers qui se trouvent sur le Bureau (à ne pas confondre avec l'appel des fichiers qui se trouvent sur disque au moyen de la commande <find>). Ce rappel, donc, se fera tout simplement avec <load0>9, qui remet dans la Macro 0 l'ordre stocké dans la Mémoire n°9, suivi de ⌘-0. C'est précisément ce que

vous pouvez voir au début de la macro ⌘-A, celle qui vous envoie dans ADRESSES.DEMO choisir votre adresse... juste après un <menu><resume> déjà familier, suivi d'un <swap> qui l'est moins. Je vais y revenir dans un instant, mais voyons d'abord ce que fait la routine principale ⌘-£.

⌘-£ commence par l'instruction de base <OA-Q><esc>, qui est la voie directe — où que l'on se trouve dans AppleWorks — pour aller au Menu principal. On crée un fichier nommé LETTRE, on le met en page (c'est le moment d'insérer votre nom et votre adresse pour l'en-tête), on place le curseur à l'endroit où devra s'écrire l'adresse du destinataire, et l'on se branche sur <SA-A>. Donc les instructions <menu> <resume> qui suivent préparent la Macro 0 à revenir au fichier LETTRE ; et comme nous allons avoir besoin de cette même Macro 0 pour aller au fichier ADRESSES.DEMO, nous allons provisoirement permuter, au moyen de <swap> le contenu actuel de la Macro 0 et celui d'une mémoire spéciale (voir manuel français de SMW, page 26). L'ordre de revenir au fichier LETTRE étant ainsi provisoirement mis à l'abri, nous pouvons charger la Macro 0 avec <load0>9 pour aller, en faisant <SA-0>, au fichier ADRESSES. DEMO. Quand la lecture de l'adresse sera terminée, nous reviendrons à LETTRE en faisant <swap><SA-0>.

## Bibliographie

Vient de paraître chez Sybex la traduction française de AppleWorks : Tips and Techniques, de Robert Ericson, qui devrait être le livre de chevet de quiconque veut exploiter à fond toutes les finesses et ressources d'AppleWorks.

Si tout cela vous déconcerte une peu, faites quelques exercices avec <menu>, <resume>, <swap> et <SA-0> sur les fichiers du Bureau ; tout s'éclairera très vite.

Où en étions nous ? Ah oui, nous débarquions dans le fichier ADRESSES.DEMO ; <OA-1> envoie au début du fichier ; avec <zoom><OA-Z> et <insert> <OA-A> on s'assure qu'on est bien en format fiche et mode recouvrement ; ajoutons un petit message pour le pauvre humain en face de l'écran, un <input> qui permettra de promener le curseur avec les flèches ou la souris, et en route pour la lecture avec <SA-L>.

La macro ⌘-L renvoie en cascade à ⌘-2, ⌘-3 et ⌘-R. Cette dernière enregistre avec des <read> une rubrique dans la Macro 0 (toujours elle !), et va à la ligne ; SA-3 emmagasine ce contenu de la Macro 0 dans une mémoire auxiliaire numérotée par le compteur <var>. Ce compteur est mis à zéro au début de SA-L, et SA-2 est la boucle d'incrémentation de <var> à concurrence de 5. Ainsi nous remplissons les six mémoires auxiliaires numérotées de 0 à 5 avec six rubriques consécutives d'une fiche adresse. Retour à la fin de SA-A, où <swap><SA-0> renvoie au fichier LETTRE. On passe à l'écriture de l'adresse avec SA-E.

Ici, même procédé que pour la lecture. On rappelle donc successivement avec <load0> <var> le contenu des mémoires auxiliaires dans la Macro 0, et on l'écrit avec SA-0 (c'est dans la routine SA-7) avec en plus, dans SA-8, un test pour sauter toute rubrique vide marquée d'un tiret... Quand c'est fait, retour à la suite de SA-£ avec l'écriture de la date, et enfin <SA-N> pour renommer le fichier LETTRE avec le premier mot de la première rubrique de l'adresse (généralement le nom du destinataire).

Et voilà. Après vous être familiarisés avec les diverses opérations constitutives de cette macro complexe vous serez mieux armés pour construire — tout en vous distrayant — vos propres applications personnelles au moyen de ce merveilleux *MECCANO* intellectuel qu'est SuperMacroWorks.

## À propos de Super Macro Works

Rappelons la signification de OA et SA :

OA = Open Apple
= Pomme ouverte
SA = Solid apple
= Pomme pleine
= option (sur le IIGS)

Un lecteur perspicace, M. Sergiuz Chadzinsky, a relevé que dans la macro Compte-Mots publiée dans le numéro 34 de Pom's, il manque la mise à zéro de la mémoire des milliers.
Il faut effectivement ajouter <save0>4 juste avant <sa-M>, dans la première macro (S:).
Cette omission est due à un ajout *in extremis* ...

(1) Pom's 34, macro-commandes AND, OR et Round pour le tableur et Compte-Mots pour le traitement de textes.

*Un correspondant pose une question qui en tourmente plusieurs autres : peut-on recopier une case valeur, définie par une formule dans un tableau, soit ailleurs dans le même tableau, soit dans un tableau différent, en récupérant la valeur et non la formule (qui évidemment n'a plus aucun sens dans un nouveau tableau).*

*Réponse : Oui, SuperMacroWorks vous donne le moyen de le faire en une seule macro. Mais... réponse dans le prochain numéro. D'ici à là, à vos claviers, et bonne chance !*

## Fichier ADRESSES.DEMO
### Fichier Base de données AppleWorks

Ce fichier figure sur la disquette Pom's et sert de démonstration pour l'utilisation de la macro-commande ADRESSAGE.
Une fiche se présente ainsi :

```
NOM & PRENOM: POM.S Editions MEV
N° et RUE: 12, rue d'Anjou
LOCALITE: -
CODE POSTAL: 78000
VILLE: VERSAILLES
PAYS: -
```

**Règles à observer pour la constitution du fichier ADRESSES.DEMO**

1 En cas de rubrique vide, mettez un tiret (nécessaire pour le bon fonctionnement de la sous-routine 8 de la macro-commande ADRESSAGE)
2 La rubrique nom et prénom ne doit contenir que lettres, chiffres, espaces ou points, à l'exclusion de tout autre signe (ainsi POM'S sera, pour les besoins de la cause : POM.S) ; sinon, la sous-routine N serait en difficulté (essayez...)
3 Chaque fiche peut avoir des rubriques supplémentaires (téléphone, remarques, etc. qui seront sans influence sur l'adressage)
4 Si en cours de macro, emportés par l'élan de votre souris, vous tombez en fin de fichier sur l'annonce fatidique "Dépassement...Voulez-vous vraiment faire cela ? Non, Oui", soyez Normand, ne répondez ni Oui, ni Non, faites Escape...
5 Sélection et tri sont impossibles en cours de macro ; prenez les devants, sélectionnez d'abord, lancez la macro ensuite.


**InterPom's**

**version 2.0**

Des transmissions intelligentes entre :
**Apple//**™
et/ou **Macintosh**™
et/ou **IBM**®

Pour transférer tous vos fichiers en local ou à distance, renseignements, documentation :
Pom's (1) 39 51 24 43



## Pom_Link 3.1

Pour réaliser un serveur Minitel avec un Apple // ou un Macintosh, il faut un outil — Pom_Link 3.1 est un ensemble complet d'instructions télématiques qui s'ajoutent à l'Applesoft ou au MSBasic — Fonctions graphiques évoluées — routines de saisie (y compris numérique et mode 'secret') contrôlées — détection de sonnerie — détection de la déconnexion à tous niveaux — contrôle du curseur — messages d'erreur et d'alerte — recodage des caractères — contrôle des touches de fonctions — interrogation du Minitel servi — contrôle de la durée de communication — mode téléinformatique — allocation de délais de réponse — Gestion des gris etc. — Prix 450,00 F (carte Joker 400,00 F) — bon de commande page 71 —

**Le Basic télématique des Apple // et Macintosh**


## Fichier ADRESSAGE
### Macro-commande pour SuperMacroWorks

*Si vous n'avez pas la disquette Pom's, saisissez en traitement de textes ce fichier (Seule la partie allant de START à END — y compris — est active).*

```
*************************************************
  APPLEWORKS 1.4              SuperMacroWorks
              ADRESSAGE AUTOMATIQUE
              =====================
Grâce aux macros de ce fichier, vous pourrez faire
un adressage automatique de votre courrier, à
partir d'un fichier d'adresses en Base de données.

Les seules opérations préalables sont :
1  Compiler ce fichier ADRESSAGE avec ⌘-=
2  Appeler à l'écran le fichier ADRESSES.DEMO, et
   faire ⌘-C pour chaîner.

Ensuite, faire ⌘-£ : la création de la lettre,
l'écriture de l'adresse, l'insertion de la date
sont entièrement automatiques. Pour chaque
nouvelle adresse il est créé un nouveau
fichier-lettre différent. On peut répéter ⌘-£
à volonté tant qu'il y a de la place sur le Bureau
(12 fichiers).

*************************************************
          Dimitri  Geystor * Fév. 1988

START

C:<all><menu><resume><save0>9<msg>'--Chainage
```

```
enregistré - Faites sa-£ pour exécuter la macro
d°adressage --'!

Marque le fichier ADRESSES.DEMO pour que la macro
puisse le retrouver
-----------------------------------------------------
£:<all><oa-Q><esc><rtn>3<rtn><rtn>LETTRE<rtn><oa-o
>mh<Rtn>1<Rtn>mb<Rtn>1<Rtn>md<rtn>0<rtn><Esc>Votre
nom et prénom<Rtn>Votre adresse<Rtn>Code postal et
localité<Rtn><Rtn><Rtn><Rtn><oa-O>MG<Rtn>4,5<Rtn><
Esc><sa-A><rtn><rtn><rtn><rtn><sa-D><rtn><rtn><oa-
O>mg<rtn>0<rtn>md<rtn>1<rtn><esc><rtn><oa-N><oa-y>
<sa-N><rtn><msg>' ---- PRET !  Vous pouvez taper
votre lettre ...----'!
                                  Routine principale

A:<all><menu><resume><swap><load0>9<sa-0><oa-1><zo
om><oa-Z><insert><oa-A><M sg>'-- Choisissez une
adresse (curseur début NOM), puis RETURN pour la
suite---'<Input><sa-L><swap><sa-0><sa-E>!
                        Envoi au fichier ADRESSES.DEMO
-----------------------------------------------------
Sous-routines :
==============

R:<all><read><read><read><read><read><read><read><
read><read><read><read><read><read><read><read><re
ad><read><read><read><read><read><read><read><read
><read><read><read><read><read><read><read><read>!

Enregistre une rubrique dans la Macro 0 (35
caractères max.)

L:<all><var=>0<sa-2>!                    )
                                         )Lecture
2:<all><sa-3><incvar><varnot>5<Rpt>!     )de l'a-
                                         )dresse
3:<all><sa-R><Rtn><save0><var>!          )

E:<all><insert><oa-A><var=>0<sa-6>!      )
                                         )
6:<all><sa-7><incvar><varnot>5<Rpt>!     )Ecriture
                                         )de l'a-
7:<all><load0><var><sa-0><Rtn><Up><sa-8>!)dresse
                                         )
8:<all><Ifnot>-<Down>!                   )

-----------------------------------------------------
D:<all>Le
<date><back><Del><back><oa-D>i<oa-Left><Rtn><oa-Ri
ght><oa-Right><Rtn><lc><a head><ahead>!

Ecrit la date
-----------------------------------------------------
N:<all><load0>0<sa-0><back><back><back><ahead><oa-
Y>!

Renomme le fichier LETTRE d'après le nom relevé
dans l'adresse
-----------------------------------------------------
S:<all><oa-Q><esc>4<rtn><up><up><up><up><up><up><u
p><up><up><up><up><right><down><right><down><right
><down><right><down><right><down><right><down><rig
ht><down><right><down><right><down><right><down><r
ight><down><right><down>< rtn><msg>'-Vide le
Bureau, avec sauvegarde manuelle des fichiers
nouveaux ou changés-'!

Fait le ménage sur le Bureau. Les fichiers déjà
sauvegardés sont enlevés d'office, les fichiers
nouveaux ou changés doivent être sauvegardés
manuellement.
-----------------------------------------------------
END
```

➡ Suite de la page 20

```
27E0:22 00 F5 02 00 9E 02 22 00 F5 02 F0 A8 02 4F 00
27F0:F5 02 00 AB 02 4F 00 F5 02 F0 B5 02 7C 00 F5 02
2800:00 B8 02 7C 00 F5 02 00 CF 02 0F 06 F5 02 F0 D2
2810:02 A4 00 F5 02 00 D5 02 A4 00 F5 02 F0 E4 02 A7
2820:00 F5 02 00 E7 02 A7 00 F5 02 00 F4 02 0F 06 F5
2830:02 F0 F7 02 D3 00 F5 02 00 FA 02 D3 00 F5 02 F0
2840:08 03 02 01 F5 02 00 0B 03 20 02 F5 02 00 0F 03
2850:04 02 F5 02 00 12 03 0F 06 F5 02 00 24 03 3A 02
2860:F5 02 00 27 03 97 05 F5 02 00 2A 03 3A 02 F5 02
2870:00 2F 03 3C 02 F5 02 00 35 03 9A 05 F5 02 00 38
2880:03 3E 02 F5 02 00 3E 03 9A 05 F5 02 00 44 03 B1
2890:05 F5 02 00 4A 03 B1 05 F5 02 00 4D 03 3A 02 F5
28A0:02 00 61 03 2E 04 F5 02 00 65 03 2B 04 F5 02 F0
28B0:6B 03 2A 06 F5 02 00 6E 03 2A 06 F5 02 00 78 03
28C0:6A 04 F5 02 00 7E 03 0F 06 F5 02 00 E4 03 46 02
28D0:F5 02 00 E7 03 3E 02 F5 01 F0 ED 03 3C 02 F5 01
28E0:F0 EE 03 44 02 F5 02 00 F0 03 3C 02 F5 02 00 F4
28F0:03 44 02 F5 02 00 FE 03 00 05 F5 02 00 0D 04 60
2900:05 F5 02 00 1C 04 3C 02 F5 02 00 1F 04 44 02 F5
2910:01 F0 25 04 3C 02 F5 01 F0 26 04 44 02 F5 02 00
2920:3D 04 44 02 F5 02 00 42 04 05 05 F5 02 00 45 04
2930:44 02 F5 02 00 4A 04 97 05 F5 02 00 4D 04 3A 02
2940:F5 02 00 50 04 3A 02 F5 02 00 58 04 3A 02 F5 02
2950:00 5D 04 97 05 F5 02 00 60 04 3A 02 F5 02 00 68
2960:04 3A 02 F5 02 F0 6B 04 1C 00 F5 02 00 6E 04 1C
2970:00 F5 02 00 78 04 97 05 F5 02 F0 7B 04 1F 00 F5
2980:02 00 7E 04 1F 00 F5 02 00 8C 04 3A 02 F5 02 00
2990:91 04 00 05 F5 02 00 A0 04 3C 02 F5 02 00 A6 04
29A0:3C 02 F5 02 00 AA 04 3C 02 F5 02 00 B0 04 9A 05
29B0:F5 02 00 BC 04 3C 02 F5 02 00 BF 04 3C 02 F5 02
29C0:00 C2 04 32 02 F5 02 00 C7 04 3C 02 F5 02 00 CC
29D0:04 3C 02 F5 02 00 D1 04 32 02 F5 02 00 D5 04 3C
29E0:02 F5 02 00 D8 04 B1 05 F5 02 00 DB 04 60 05 F5
29F0:02 00 01 05 3C 02 F5 02 00 21 05 60 05 F5 02 00
2A00:3F 05 16 05 F5 02 00 43 05 13 05 F5 02 F0 49 05
2A10:95 06 F5 02 00 4C 05 95 06 F5 02 00 61 05 E0 04
2A20:F5 02 00 64 05 3E 02 F5 02 00 6E 05 B2 06 F5 02
2A30:00 71 05 40 02 F5 02 00 74 05 AA 06 F5 02 00 77
2A40:05 42 02 F5 02 00 7A 05 AE 06 F5 02 00 80 05 99
2A50:06 F5 02 00 85 05 4E 02 F5 02 00 88 05 40 02 F5
2A60:02 00 8C 05 42 02 F5 02 00 8F 05 4E 02 F5 02 00
2A70:94 05 99 06 F5 02 00 98 05 3A 02 F5 02 F0 9D 05
2A80:47 01 F5 02 00 A0 05 10 02 F5 02 00 A4 05 F0 01
2A90:F5 02 00 A7 05 0F 06 F5 02 00 B5 05 98 01 F5 02
2AA0:00 B8 05 3C 02 F5 02 00 C0 05 4C 02 F5 02 00 D3
2AB0:05 98 01 F5 02 00 D7 05 FC 01 F5 02 00 DA 05 0F
2AC0:06 F5 02 00 ED 05 95 01 F5 02 00 F1 05 95 01 F5
2AD0:02 00 FF 05 95 01 F5 02 F0 02 06 93 01 F5 02 00
2AE0:05 06 93 01 F5 02 00 2B 06 E0 04 F5 02 F0 34 06
2AF0:B1 01 F5 03 00 37 06 26 02 F5 03 00 3C 06 0C 02
2B00:F5 02 00 40 06 0F 06 F5 02 F0 4A 06 9C 01 F5 02
2B10:00 4D 06 9C 01 F5 02 F0 57 06 9A 01 F5 02 00 5A
2B20:06 9A 01 F5 03 00 64 06 A4 01 F5 03 00 6B 06 A4
2B30:01 F5 03 00 6F 06 0E 02 F5 03 00 74 06 4C 02 F5
2B40:02 00 7D 06 0F 06 F5 02 F0 80 06 A4 01 F5 02 00
2B50:83 06 A4 01 F5 02 00 96 06 E0 04 F5 03 00 A6 06
2B60:99 06 F5 03 00 B9 06 B2 06 F5 03 00 C0 06 BD 06
2B70:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2B80:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2B90:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2BA0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2BB0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2BC0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2BD0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2BE0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2BF0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

# ProDOS : Petit guide de l'utilisateur

Pierre Demblon

Une des grandes forces de ProDOS est de pouvoir gérer sans problème toutes sortes de mémoires de masse : qu'il s'agisse de nos bons vieux lecteurs 5'25, des disques 3'5, d'un disque dur, d'un disque-mémoire (Ramdisk), de la carte Speedisk, et bientôt d'un lecteur de 'Compact Disk', que la carte interface soit de type "Smartport" ou "SCSI", l'utilisateur n'a jamais affaire qu'à des volumes, des catalogues, des sous-catalogues et des fichiers. Le programmeur, lui, a affaire à une interface en langage machine elle aussi très simple et normalisée, et n'a jamais à s'occuper de la gestion directe des lecteurs.

La conséquence bien agréable pour l'utilisateur, c'est l'abondance des applications et des mémoires de masse, donc la nécessité de moyens simples et rapides pour copier, effacer, formater disques et fichiers, passer d'une application à une autre, faire des 'backups' (des doubles).

J'ai pu constater que beaucoup d'utilisateurs perdent leur temps et leur sérénité parce qu'ils se contentent d'utiliser les utilitaires remis avec leur machine (Finder, MouseDesk, Utilitaires Système). Le présent article vise à faire l'étude à peu près complète (d'où sa longueur...) des programmes du commerce indispensables à l'utilisateur de ProDOS.

## Prenez vos précautions !

Commençons par quelques mises en garde indispensables pour éviter les désastres (ces désastres, bien entendu, me sont tous arrivés : si vous n'y prenez pas garde, ils vous arriveront fatalement à vous aussi !).

### Attention à votre ProDOS

N'utilisez pas n'importe lequel, simplement parce que c'est celui-là qui est sur la disquette de l'application que vous utilisez. Il est indispensable de vous faire vous-même un disque système ProDOS, votre disque de démarrage, adapté à votre configuration, contenant le bon ProDOS, vos drivers de disques Ram, vos accessoires de bureau, un sélecteur de programmes, un gestionnaire de fichiers, etc., et épuré de tout ce qui ne vous est pas utile (polices de caractères, AppleTalk, drivers de LaserWriter...). C'est à partir du ProDOS de cette disquette que vous lancerez vos applications, et non à partir du ProDOS fourni sur la disquette de l'application. En fonction de ce que j'ai dit ci-dessus concernant l'interface pour le programmeur, toute application ProDOS doit être compatible avec les nouvelles versions de ProDOS. Si elle ne l'est pas, procurez-vous la version la plus récente de cette application. S'il n'y a pas de version plus récente... n'utilisez plus cette application : vous courez au désastre !

Le mot "ProDOS" renvoie d'une part au ProDOS employé sur tous les Apple//, appelé aujourd'hui ProDOS 8, et d'autre part au système d'exploitation 16 bits du IIGS, appelé ProDOS 16. Ces deux systèmes gèrent les disques de la même façon. La seule version de ProDOS 8 qu'il faut utiliser est la version 1.4, datée d'avril 87. Toutes les versions antérieures sont affectées de 'bugs' cruels (destruction de la piste 0...), ou de manques et d'incompatibilités graves (inhibition des interruptions sur le IIGS, c'est-à-dire plus de tableau de bord, calendrier qui s'arrête au 31 décembre 1987, difficultés à ressusciter les fichiers détruits...).

Il faut quelques années à un système d'exploitation pour être bien rodé, c'est inévitable. Pour ProDOS 8, la maturité est atteinte. Pour ProDOS 16, ce n'est pas encore le cas : la seule version à conseiller est la version 2.0... qui n'est pas encore réalisée par Apple. En attendant, il faut se débrouiller et jongler avec les diverses versions, en fonction de la version du GS et des applications utilisées (mais sachez que les versions trop primitives du système ProDOS 16 n'acceptent pas le ProDOS 8 version 1.4).

Si vous avez un GS, vous avez déjà le ProDOS 8 version 1.4, mais vous ne le savez peut-être pas. Ce qui est appelé "ProDOS" sur vos disquettes 'système' est seulement un utilitaire de chargement du système, le ProDOS 8 se trouve dans le sous-catalogue "System", sous le nom "P8". Il suffit de formater une disquette, d'y recopier "P8" et de le renommer "ProDOS" pour avoir un disque qui démarre automatiquement avec le bon ProDOS 8. Pour ceux qui ont un Apple 8 bits, votre concessionnaire doit faire cette manipulation pour vous. S'il ne sait pas la faire, dénoncez-le à Apple France et adressez-vous à un ami avec un IIGS.

### Attention à vos applications

Le support sur lequel elles vous sont livrées (disquettes 5'25 ou 3'5) est seulement un intermédiaire provisoire pour la transmission. C'est à vous de recopier les fichiers de cette application sur le disque désiré, dans le sous-catalogue désiré, et de configurer cette application non seulement pour votre imprimante et vos ports, mais pour qu'elle aille chercher et sauver ses données là où vous lui direz. Beaucoup d'excellentes applications 8 bits, à commencer par AppleWorks, utilisent la méthode du relais-disque (overlay) pour dépasser la limite des 128Ko : elles gagnent énormément en vitesse si on les installe sur un disque Ram.

Cela implique des exigences précises à adresser à toutes les applications ProDOS que vous utilisez : si une application n'est pas conforme à ces exigences, encore une fois, ne l'utilisez pas !

① une application ProDOS ne doit jamais être liée au disque sur lequel elle est livrée, elle doit être transportable sur le support décidé par vous. Cela exclut bien évidemment la protection contre la copie, puisque précisément vous devez copier l'application avant de l'utiliser. Mais cela exclut aussi les programmes qui exigent de travailler à partir d'un catalogue ou sous-catalogue déterminé, et ne sont pas configurables. En revanche cela inclut l'essentiel des bons programmes sous ProDOS dont... les disquettes et programmes de Pom's.

② une application ProDOS doit toujours avoir une option "Quit" qui renvoie au ProDOS, c'est-à-dire en fait à votre sélecteur d'applications, on ne doit pas être obligé de redémarrer tout. Qui a chargé une seule fois le ProDOS 16 comprendra pourquoi...

③ une application ProDOS doit autoriser les interruptions, c'est-à-dire l'accès au tableau de bord, et permettre l'aller-retour avec vos accessoires de bureau.

Cette liste peut paraître bien exigeante : mais je répète que les bons programmes sous ProDOS s'y conforment effectivement, d'AppleWorks à Publish-It en passant par Graphic Writer pour GS et par les applications qui font l'objet du présent article. Et sachez que Copy // Plus version 8 permet de faire une version 'disque dur", c'est-à-dire conforme à ces exigences, des quelques programmes GS qui ne s'y conforment pas.

### Attention au bloc 2

Quel que soit le disque que vous utilisez habituellement, il y a un bloc du disque qui est toujours lu plusieurs fois, et la plupart du temps changé, à chaque accès disque. C'est le bloc 2, le premier secteur du catalogue principal. Ce bloc est même lu quand vous n'utilisez pas le disque : il suffit que celui-ci soit physiquement présent dans un lecteur pour qu'à chaque commande *On Line* de ProDOS (et elles sont fréquentes : combien de fois par heure voyez-vous et entendez-vous les lecteurs s'allumer les uns après les autres pour un très court instant ?), le bloc 2 soit lu.

La conséquence est inévitable : s'il y a un disque que vous utilisez habituellement, ou simplement qui reste en ligne en permanence, quelle que soit sa taille ou sa nature, y compris disque dur ou disque Ram, ce bloc sera fatalement abîmé un jour sur ce disque. La question n'est pas de savoir s'il le sera, mais quand il le sera. Il est impossible, quels que soient la perfection du système d'exploitation et la fiabilité du matériel, qu'un même bloc du disque soit lu et écrit des dizaines de milliers de fois sans un pépin. Utiliser régulièrement un disque sous ProDOS sans aucune précaution contre la destruction du bloc 2, c'est partir pour le Paris-Dakar sans roue de secours.

La question est donc de savoir quelles précautions prendre maintenant pour pouvoir se tirer sans trop de mal de ce pépin qui arrivera nécessairement un jour. La première réponse, ce sont les 'backups' : avoir systématiquement un double de ce qui est un tant soit peu important. Je n'insiste pas sur cette règle d'or, qui est le B.A. – BA de l'informatique.

Mais il y a d'autres réponses, qu'on connaît moins. On peut aisément sur un disque ProDOS retrouver un fichier perdu à cause de l'effacement du catalogue, mais on ne pourra pas savoir le nom, le type du fichier, son adresse éventuelle de chargement, sa longueur, tous renseignements indispensables pour rendre utilisable le fichier ressuscité qui sont inscrits uniquement dans le catalogue ou sous-catalogue. Il n'y a qu'une exception à cela : si le fichier est un sous-catalogue.

Le bloc 2 contient le nom de volume du disque ProDOS, et les 12 premiers noms et pointeurs de fichiers. En conséquence, tout catalogue d'un disque de travail ProDOS devrait commencer par 12 noms de sous-catalogues : non seulement un fichier qui est dans un sous-catalogue ne sera pas atteint par la destruction du bloc 2, mais il est extrêmement facile aux utilitaires de résurrection des disques abîmés de retrouver le sous-catalogue entier sur le disque, y compris son nom.

Pour la même raison, on a tout intérêt à avoir quelque part dans un sous-catalogue du disque un fichier texte où soit capturé le catalogue complet du disque, pour pouvoir reconstituer ce catalogue le jour où le bloc 2 sera abîmé. Cela est parfaitement possible avec le programme **INFO.DESK** inclus dans Prosel (voir ci-dessous).

Enfin, il est sans doute nécessaire d'avoir un double du bloc 2 lui-même. La solution que je pratique est la suivante : dans un de mes sous-catalogues, je crée un fichier appelé "Bloc.2", d'un type exotique, de longueur 512 :

```
CREATE BLOC.2, TIVR.↵
BSAVE BLOC.2, TIVR, A$800, L512.↵
LOCK BLOC.2.↵
```

Je lance INFO.DESK de Prosel pour savoir quel bloc ProDOS a réservé à ce fichier, qui n'occupe qu'un seul bloc. Puis, avec un éditeur de blocs, je recopie le bloc 2 sur le bloc en question. Le jour où mon bloc 2 est détruit, je n'ai plus qu'à recopier ce bloc. Cela peut se faire en une seule ligne de programme avec les commandes RAMLOAD et RAMSAVE du programme UNIRAM (voir ci-dessous).

## Quels outils ?

De quoi avons-nous besoin exactement pour gérer nos disques ProDOS ?

① un sélecteur de programmes, c'est ce qui vous permet de passer d'une application à une autre sans avoir à faire de redémarrage. On peut les classer suivant deux critères : j'appellerai *sélecteurs ouverts* ceux qui, à chaque fois qu'on fait appel à eux, cherchent sur tous les disques en ligne les programmes qu'il peuvent lancer, et vous affichent ce choix. J'appellerai *sélecteurs fixes* ceux qui contiennent une liste d'applications et des chemins d'accès à ces applications, liste que vous leur avez préalablement déclarée (et que vous pouvez bien entendu changer).

② un bon "Filer" et de bons copieurs de disques, qui permettent rapidement et aisément de créer, détruire, renommer, copier les fichiers, formater et dupliquer les disques. ProDOS fait tout ça très bien, mais il ne comprend que le langage machine. Il nous faut donc un bon programme d'interface entre lui et nous pour réaliser toutes ces opérations en direct.

③ cela ne suffit encore pas : en effet, nous aurons souvent besoin d'automatiser le travail du Filer. Par exemple, quand on a un disque Ram de taille suffisante, on gagne un temps fou à installer le système ProDOS et les applications les plus couramment utilisées sur ce disque Ram, et à lancer le système à partir de lui. Mais comme tout cela disparaît quand on éteint la machine (sauf pour les heureux propriétaires de la carte Speedisk), il faut transférer pas mal de fichiers du disque et créer pas mal de sous-catalogues. Il faut pour cela écrire un programme, donc utiliser un langage qui serve d'interpréteur au système d'exploitation.

④ nous l'avons vu, il est indispensable de faire des 'backups' de

tous les disques et programmes que nous utilisons souvent, et bien entendu de tout ce qui réside dans un Ramdisk. Mais comme nous n'avons pas souvent un deuxième disque dur, un deuxième lecteur 3'5, etc., il nous faut des utilitaires qui permettent (puisque ProDOS le permet), de faire un double du disque Ram sur le dur, un double du dur sur des disquettes 3'5, etc., et cela en fonction de notre configuration précise, et sans perdre trop de temps. Puisque le temps, c'est de l'argent, et même si ProDOS est un système d'exploitation assez rapide, comme nous gérons l'espace disque en Méga-octets, tout ce qui pourra accélérer le travail sera le bienvenu.

⑤ quelles que soient nos précautions et nos backups, un jour il nous faudra réparer des blocs, des fichiers et des pistes endommagés. Cela nous prendra du temps, mais moins que de reconstituer les données perdues. Il nous faut donc aussi prévoir les utilitaires d'accès direct au disque, de réparation et de reconstitutions de catalogues et de disques.

Bien entendu, aucun programme ne peut remplir ces six fonctions à la fois, mais la plupart en remplissent plusieurs. C'est donc par groupe de programmes, plutôt que par fonctions, que je les présenterai.

## ENTER PATHNAME OF NEXT APPLICATION

Le message ci-dessus est responsable de nombre d'angoisses et de crises de nerfs. C'est ce que ProDOS 8 a à nous proposer en guise de sélecteur d'applications : Apple a volontairement fait le plus dépouillé possible, pour permettre aux développeurs de faire des programmes qui se logent dans le ProDOS 8 lui-même, à la place de cet infâme message. C'est ainsi que nous avons aujourd'hui, sans occuper un octet de plus en mémoire, et la plupart du temps sans occuper non plus plus de place sur le disque, d'excellents mini-sélecteurs indispensables à tout utilisateur de ProDOS 8.

Celui dont je me sers quotidiennement est le **Bird's Better Bye** (BBB) d'**Alan Bird** de **Beagle** Bros. Il est tranquillement logé dans le fichier ProDOS, et on ne sait même pas qu'il est là tant qu'on ne fait pas appel au Quit (en tapant "Bye" sous Basic par exemple). Mais dès qu'on sort d'une application, l'écran affiche la liste des fichiers 'lançables' et des sous-catalogues du disque. On se déplace dans cette liste avec les touches ↑↓, il suffit d'appuyer sur 'Return' pour lancer l'application choisie, ou retrouver une nouvelle liste si vous avez choisi un sous-catalogue. La touche 'Escape' fait passer au disque suivant.

Ce sélecteur est extrêmement rapide et efficace, surtout si vous avez un disque principal sur lequel résident la plupart de vos applications. C'est celui qu'il vous sera le plus facile de trouver, car c'est celui que Pom's diffuse : en prime, Pom's vous offre d'ailleurs sur la même disquette une application intéressante : SuperMacroWorks pour AppleWorks. C'est ce programme qu'il faut commander page 74 pour avoir le BBB en français, caché dans le ProDOS (Version 1.4) de la disquette.

Un autre sélecteur ouvert intégré au ProDOS 8 est celui réalisé par **Glen Bredon**. Lui aussi est invisible au démarrage et ne se manifeste que par un appel au "Quit". Il fonctionne un peu différemment du BBB : il commence par cataloguer tous les disques en ligne, et vous propose de choisir votre volume avec les ↑↓ et 'Return'. Quand ce choix est fait, il affiche les fichiers 'lançables' du catalogue en haut de l'écran, et en bas la liste des sous-catalogues contenus dans ce catalogue. Les ↑↓ et 'Return' permettent de lancer une application, la touche 'Tab' permet la navette entre les sous-catalogues et le catalogue principal. Le plus intéressant dans ce sélecteur est qu'il dispose d'une commande 'Pop' : quand vous êtes dans un sous-catalogue, la touche 'Escape' vous permet de remonter d'un cran dans la hiérarchie des sous-catalogues et de lister les applications et sous-catalogues du niveau supérieur. Il est donc particulièrement utile à ceux qui ont un disque dur très structuré, avec beaucoup de niveaux de sous-catalogues.

Vous pouvez trouver ce sélecteur vendu par **Roger Wagner** avec une intéressante application en prime : il s'agit de... Merlin 8/16 (voir à ce sujet la chronique "Apple // For Ever" de J.Y. Bourdin dans Pom's 34). Pour ce sélecteur comme pour le BBB, il est parfois gênant de ne pas les avoir en fichier séparé, si vous voulez passer de l'un à l'autre ou les mettre dans le futur ProDOS 8 Version 1.5 ou 2.0. Pour celui de Merlin, il est d'ailleurs nécessaire de le sortir de son ProDOS (Version 1.3) pour le mettre dans la version 1.4. Voici comment s'y prendre :

```
BLOAD/MERLIN/PRODOS, A$2000, TSYS.↵
BSAVE/MON.DISQUE/QUIT, A$5900, L$300.↵
BLOAD/MON.DISQUE/PRODOS, A$2000, TSYS.↵
BLOAD/MON.DISQUE/QUIT, A$5900.↵
BSAVE/MON.DISQUE/PRODOS, A$2000, TSYS.↵
```

Vous trouverez aussi ce sélecteur (avec sa documentation) dans l'indispensable Prosel (voir ci-dessous) sous la forme d'un fichier appelé "Select.System".

Je passerai rapidement sur le sélecteur diffusé par **Checkmate Technology**, appelé **SQUIRT**, qui est lui aussi ouvert, s'intègre aussi à ProDOS 8, mais se présente en fichier sur disque : je n'aime pas la formule du choix avec des lettres A/B/C/D etc. Son avantage est son prix : si vous avez la carte **Multiram** ou le programme **Proterm** de **Checkmate**, vous l'avez déjà. Sinon, il est diffusé en 'Shareware' : vous l'obtenez pour $3.50 du **Big Red Computer Club**, ou gratuitement d'un de vos amis, et s'il vous plaît, vous envoyez $7.50 à l'éditeur pour avoir le droit de l'utiliser. Pour l'intégrer définitivement à votre ProDOS 8, faites la même chose que ci-dessus, sauf pour le début. Tapez :

```
BLOAD SQUIRT.SYSTEM, A$5700, TSYS.↵
```

le reste comme ci-dessus.

*Pour les techniciens : vous avez déjà compris qu'il y a à l'intérieur même de ProDOS 8 trois pages réservées au code du QUIT dans lesquelles on peut mettre son propre code. Je vous recommande l'article de Sandy Mossberg dans Nibble de septembre 87, qui désassemble le code Quit du ProDOS, et offre un source Merlin pour faire vous-même votre Quit personnalisé.*

Si vous voulez démarrer directement sur ces sélecteurs intégrés à ProDOS sans passer par Basic.System et "Bye", il suffit de mettre sur le disque, comme premier fichier portant le nom XXX.SYSTEM, un fichier BYE.SYSTEM qui se contente de faire un appel au Quit du ProDOS (plus exactement, mettez-le en second, juste après SETUP.SYSTEM : voir "Apple // For Ever" dans ce numéro). Voici comment créer ce fichier :

```
CREATE BYE.SYSTEM, TSYS.↵
CALL-151.↵
2000: 20 00 BF 65 06 20 04 00 00 00 00 00 00 00 D0 00
      00 00 00.↵
BSAVE BYE.SYSTEM, A$2000, L$13, TSYS.↵
```

## Ceux qui font beaucoup : MOUSEFILER et RUNRUN

**Mousefiler**, d'**Éric Lapuyade** ne doit pas, parce qu'il est diffusé par **Froggy Software**, être pris pour un jeu : s'il peut

parfois sembler curieux de se déplacer avec la souris dans un éditeur de blocs, il reconnaît aussi le clavier, marche avec les ⌘⌥. C'est un très bon sélecteur fixe, doublé d'un Filer très honnête et d'un éditeur de blocs. Il exécute la plupart des fonctions disque du ProDOS 8, et est agréable à utiliser.

**RunRun**, de **Pinpoint Publishing**, c'est le Finder pour //e et //c. Sélecteur fixe doté d'une interface très agréable (fenêtres réglables), il peut cataloguer les disques, lancer les applications 8 bits. Il est très aisé d'ajouter ou ôter une application de la liste. Son principal intérêt est de permettre de lancer les accessoires Pinpoint comme accessoires de RunRun, sans passer par Pinpoint ni AppleWorks. Pour un //e ou un //c, c'est le moyen de rivaliser avec le IIGS.

Une remarque tout de suite : n'achetez surtout pas directement RunRun. Pour le même prix, achetez l'excellent **Pinpoint Toolkit**. Vous aurez à la fois RunRun, Pinpoint (pas les accessoires pour AppleWorks, mais Pinpoint), tout un paquet d'accessoires inédits et très précieux pour Pinpoint (calculette hexadécimale par exemple), tout ce qu'il faut pour mettre Pinpoint dans toutes les applications possibles, et surtout la documentation, les outils et les sources nécessaires pour faire vous-même vos accessoires Pinpoint, en assembleur, en Kyan Pascal ou en Micol Basic. Il est bien rare de voir un éditeur de programmes révéler tous ses secrets et vous fournir ses sources : bravo Pinpoint !

Une seconde remarque : les accessoires Pinpoint sont concurrencés, sur IIGS, par les accessoires IIGS. La combinaison des applications **Time Out** de **Beagle** pour AppleWorks et des accessoires du IIGS m'y semble bien plus intéressante.

Enfin, si vous aimez l'interface AppleWorks (ou que vous en avez simplement l'habitude), le logiciel **Interface** de **Star Development** vous offre un sélecteur fixe avec une calculette et des fonctions de copies de fichiers qui vous fera croire que vous n'avez pas quitté AppleWorks.

## Celui qui fait tout : PROSEL

Terminons par le meilleur, l'indispensable, celui que tout le monde doit avoir quelle que soit sa configuration : il s'agit de **Prosel**, réalisé et vendu directement par **Glen Bredon** (oui, l'auteur de Merlin). Au départ, c'est une suite d'utilitaires réalisés par lui pour gérer son disque dur Sider. Dans l'état actuel des choses, c'est à mon avis le meilleur programme de gestion des mémoires de masse qui existe sur tout ordinateur individuel. Ce programme est une suite d'utilitaires impressionnante, qui permet de faire tout ce qu'il est possible de faire sur tout disque ProDOS et sur tout Apple //.

Le sélecteur de programmes lui-même, qui donne son nom à l'ensemble, est impeccable : il permet (dans les dernières versions de Prosel) de lancer aussi bien les applications ProDOS 8 que ProDOS 16, de lancer un fichier Applesoft, de lancer un fichier DOS 3.3, d'éditer la liste d'applications sans sortir du sélecteur ; il marche avec la souris comme avec le clavier, fonctionne en 160 colonnes avec la carte Ultraterm, il permet aussi de lancer les accessoires Pinpoint, il trouve et corrige les 'bugs' de vos Ramdisks, bref, c'est indiscutablement le meilleur.

Mon stock de superlatifs étant épuisé, je ne mettrai pas de qualificatifs aux différents utilitaires que Prosel nous offre en plus du sélecteur. Ils sont tous indispensables, je me contenterai d'en faire la liste.

— **Cat Doctor** est un Filer complet et efficace ;
— **Copy** copie des volumes entiers, y compris par exemple un disque 800Ko sur un disque Ram 1 Méga ;
— **Block Warden** est un excellent éditeur de blocs très avancé ;
— **Find File** trouve un fichier en cherchant lui-même dans les sous-catalogues, et vous lit ce fichier ;
— **Mr Fixit** répare les catalogues détruits ;
— **Info Desk** vous épluche les catalogues et l'occupation d'un disque sur écran, sur imprimante ou dans un fichier texte ;
— **Backup** et **Restore** permettent la copie et la restauration rapides de volumes ProDOS ;
— **Beach Comber** réorganise les fichiers sur votre disque de travail pour qu'ils y soient mieux rangés (AppleWorks par exemple, qui sauve ses fichiers en conservant l'ancienne version pour ne la détruire que quand la nouvelle est sur le disque, par mesure de sécurité, aboutit à disperser les fichiers sur la disquette, ce qui allonge les temps de chargement) ;
— Et il y a bien d'autres utilitaires dont je vous laisse la surprise...

## The Desktop Manager : des accessoires pour IIGS

Un accessoire de bureau est un programme auquel vous avez accès en permanence à l'intérieur même des applications que vous utilisez : le IIGS se charge précisément de suspendre l'application en cours, permettre le travail de l'accessoire, puis relancer l'application au point où elle avait été suspendue. Les accessoires sont l'une des raisons majeures justifiant l'achat d'un IIGS, et périment aussi bien les anciens accessoires type Pinpoint que certains des accessoires Time Out pour AppleWorks. Avoir un Filer en accessoire permet de faire toutes vos opérations sur disques sans quitter l'application en cours. Il faut bien entendu avoir chargé vos accessoires au départ : ils doivent être sur votre disque ProDOS de démarrage.

Connaissez-vous la société **On Three** ? Si oui, c'est que vous êtes le lecteur de Pom's qui a un Apple ///, car cette société est l'émanation d'un groupe d'utilisateurs de cette machine, d'où son nom. Mais il se trouve que le IIGS a des points communs avec l'Apple ///, et que cette société est dirigée par un des plus grands connaisseurs de l'Apple : **Sean Nolan**. C'est pourquoi, après le succès de l'excellent Graphics Manager, elle peut lancer une série très prometteuse pour le IIGS, appelée **TDM, The Desktop Manager**. Dans le programme de base, vous avez toutes les facilités pour installer vos accessoires de bureau, le noyau de base, plus 10 accessoires, dont certains (Printer Manager, Accessory Manager, Device Configuration, Clipboard Tools) sont indispensables. Mais la série continue avec **The Disk Manager**, qui vous permet de faire tous les travaux de gestion courante des fichiers et des disques, et le **Super Accessories #1**, qui contient entre autres un éditeur de blocs, un utilitaire pour lister les fichiers, etc.

**Attention** : ces programmes étant issus des excellents accessoires qu'On Three avait réalisés pour l'Apple ///, ce ne sont ni des NDA ni des CDA, mais une nouvelle race indépendante, dont l'interface est en texte et de type AppleWorks. Si vous exigez la souris et les fenêtres réglables, ils ne sont pas pour vous. Mais si c'est la puissance et la rapidité qui vous attirent, alors ils sont bien au-dessus de beaucoup d'accessoires-jouets qui circulent sur le IIGS.

Le seul problème avec tous ces accessoires, c'est que vous allez en considérer la moitié au moins comme indispensables (il y a aussi un **Communications Manager**...). Il va vous falloir de

la mémoire... d'autant que le plus précieux, c'est sans aucun doute le **TDM TOOLKIT** qui vous montre comment faire vous-mêmes vos accessoires, et inclut les sources, les macros, et les exemples nécessaires en assembleur. Prosel, le Desktop Manager et le TDM Toolkit, cet ensemble est une raison suffisante pour acheter un IIGS.

## L'indispensable COPY // PLUS

Il existe une application 8 bits, donc qui tourne sur tous les Apple //, qui fait vite et bien, en un seul programme, tout ce qu'on peut attendre d'un Filer. Je m'en sers tous les jours. Il s'agit du "Filer" de **Copy // Plus** de **Central Point Software**, appelé "Util.System" (je parle ici de la version la plus récente de ce programme qui soit en ma possession, la version 8.2). D'abord, ce programme copie les fichiers indifféremment en DOS 3.3 ou en ProDOS, en les convertissant automatiquement, il formate et copie les disquettes en DOS ou en ProDOS, à votre choix. Ni le Finder ni Prosel ne font cela. Il gère parfaitement les sous-catalogues, il copie en une seule passe un disque 3'5 si vous avez 800Ko de Ram libres. On le commande avec les ↑↓ et 'Return', ce qui peut sembler archaïque, mais est en fait et de très loin la formule la plus rapide.

Je ne passe jamais une heure sur ma machine sans me servir de ce programme : ses qualités de rapidité et de fiabilité sont exemplaires. On peut certes encore exprimer quelques regrets (les catalogues ne sont pas assez complets, il n'a pas de macros, il permet seulement d'alphabétiser un catalogue, mais pas d'y déplacer les fichiers à sa guise,...), mais il faut bien laisser quelque chose à ajouter pour les versions suivantes : dans son état actuel, il est absolument indispensable à tous les utilisateurs d'Apple //.

D'autant que pour le même prix, vous aurez aussi deux autres copieurs physiques (Nibble-Copiers), pour disquettes 5'25 et 3'5. Non seulement vous pouvez désormais copier physiquement les disquettes Mac sur le IIGS, mais il se charge automatiquement de mettre au standard ProDOS en les déprotégeant les quelques programmes IIGS sous ProDOS 16 qui étaient protégés. Par ailleurs ces copieurs incluent des utilitaires de réglage des lecteurs et d'édition de blocs et de pistes indispensables pour réparer les disques abîmés.

## DISKCOPY A: B: *.*

Si le sous-titre ci-dessus ne vous dit rien, réjouissez-vous : vous êtes un pur. Mais dans le monde de l'informatique, il y a aussi les "prisonniers du boulot" dont parle la chanson, qui n'ont pas le choix de leur machine, et travaillent sous des systèmes d'exploitation appelés Unix ou MS/DOS dont ils finissent par prendre tellement l'habitude qu'ils souhaitent les retrouver sur leur Apple.

C'est ainsi que s'expliquent nombre de programmes interpréteurs, de langages chargés de traduire ProDOS en instructions que vous pouvez frapper au clavier, ou regrouper en programmes, ce que, dans un autre univers, on appelle des "fichiers-batch". Sauf si vous êtes vous-même l'un de ces prisonniers, vous ne verrez guère dans ces programmes qu'un jargon inconnu pour faire des opérations disques banales.

Mentionnons-les cependant. **KIX**, de **Kyan Software** (ceux qui diffusent le très bon Kyan Pascal), vous permet par exemple de ne plus taper "CATALOG", mais "LS.", et de ne plus écrire "RENAME /DISK.A, /DISK.B", mais "MVV DISK.B DISK.A". Étonnant, non ? Surtout quand on sait que vous pouvez l'installer dans AppleWorks. Si Kix imite Unix, tout comme **ECP (Extended Command Processor)** de **Living Legend Software**, **Command.Com** de **Pinpoint Publishing** vous offre, lui, carrément la ligne de commande, et même le caractère d'appel (>) du MS/DOS. Arrêtons là.

Je n'inclurai pas dans cette liste le "shell" (ou "environnement de développement") d'APW/ORCA pour IIGS, diffusé en France par **Prim'vert-DDA**. D'abord parce qu'il ne se contente pas de singer d'autres systèmes d'exploitation, mais donne réellement des possibilités nouvelles, ne serait-ce que le moyen de faire soi-même ses commandes (d'ailleurs les 12 utilitaires supplémentaires rassemblés sous le nom d'**ORCA/M Utility #1** et diffusés par **Byteworks** incluent les sources en assembleur). C'est réellement un outil de travail pour les programmeurs. Mais en-dessous de 10 Mégas d'espace disque et de 2 Mégas de mémoire en ligne, il n'est pas question d'utiliser réellement APW, qui n'a d'ailleurs jamais prétendu être destiné aux utilisateurs : c'est l'Apple Programmer's Workshop.

```
10 PRINT CHR$(4);"COPY/DISK.A/STARTUP,
   /RAM/STARTUP"
```

La ligne ci-dessus doit vous paraitre un peu plus familière que le sous-titre précédent. Elle nous rappelle que le meilleur interpréteur pour ProDOS, nous l'avons tous déjà : il s'appelle **Basic.System**. Les "fichiers-batch", cela existe depuis des années sur Apple // sous la forme de programmes Basic, tout simplement. C'est avec un programme Basic de 10 lignes que je remplis mon disque Ram très rapidement de 400Ko de programmes et applications diverses quand j'allume ma machine.

Évidemment, si vous recopiez le sous-titre ci-dessus, votre Apple va vous dire "SYNTAX ERROR" : c'est que vous avez oublié cette possibilité extrêmement précieuse de Basic.System de rajouter des commandes à partir du disque et que vous n'avez pas ajouté la commande **Copy** de **Pom's 32**. Depuis que ProDOS existe, un grand nombre de commandes, la plupart compatibles entre elles, ont été publiées, dont beaucoup sont précisément orientées sur la gestion des disques.

La première source de ces commandes supplémentaires, vous l'avez entre les mains : c'est la revue Pom's. Voici ci-dessous une liste des commandes Basic.System publiées dans Pom's jusqu'à ce jour :

| | | |
|---|---|---|
| **INIT** | Pom's 20 | formatage disquette 5'25 |
| **TDUMP** | Pom's 20 | affichage de fichiers texte |
| **DATHEUR** | Pom's 24 | affichage de la date et l'heure |
| **SHOW** | Pom's 28 | émulation de la commande 'Mon' du DOS 3.3 |
| **INDEX** | Pom's 29 | recherche de blocs index |
| **RBLOC** | Pom's 29 | lecture d'un bloc |
| **WBLOC** | Pom's 29 | écriture d'un bloc |
| **KRUPTOS** | Pom's 29 | cryptage de fichiers |
| **COPY** | Pom's 32 | copie de fichiers |
| **XCAT** | Pom's 32 | catalogue complet (y compris dossiers) |
| **ONLINE** | Pom's 33 | liste des volumes ProDOS |

Si Pom's ne vous suffit pas, utilisez **PROCOMMAND - The Commander** de **Glen Bredon** (eh oui, encore lui), qui vous offre non seulement tout un lot de commandes Basic, mais aussi tout le nécessaire pour programmer en Applesoft sous ProDOS et gérer sans problème la double-haute résolution à partir de l'Applesoft. **L'Apple Coop** vend cet excellent

ensemble pour un prix dérisoire.
Les frères **Beagle** et **Randy Brandt** nous offrent également 18 commandes extrêmement intéressantes dans Big-U. Certaines ne quittent jamais mon Basic.System.

Si nous nous limitons strictement à la gestion des disques sous ProDOS, il faut absolument mentionner un programme indispensable que personne ne connaît : il s'agit d'**UNIRAM**, écrit et diffusé par **Steven Humpage**. J'ai acheté ce programme au départ pour une raison simple : c'est le seul à faire un Ramdisque ProDOS pour les cartes d'extension mémoire type **Saturn** ou **Legend** (les cartes ancienne mode qui font du 'bank switching' de cartes langage jusqu'à 1 Méga). Mais j'ai eu la fort agréable surprise d'y découvrir aussi un Ramdisque sous DOS et un ensemble de commandes Basic extrêmement précieuses (je ne parle ici que des commandes orientées disques, mais il y en a d'autres).

La commande **PART** permet de segmenter un fichier pour sauver par exemple en deux morceaux sur disque 5'25 un fichier de 300 blocs, la commande **ADD** permettant l'opération inverse (fusion de deux fichiers). La commande **MOUNT** permet de recopier l'intégralité d'un disque ProDOS sur un autre de taille au moins égale en recopiant seulement les blocs utilisés, elle fait la même chose que Diversi Copy (sauf le formatage des disques), mais en restant sous ProDOS et sous Basic. Les deux plus précieuses sont à mon avis les commandes **RAMSAVE** et **RAMLOAD**. La première permet de sauver dans un fichier ProDOS les blocs que vous voulez d'un disque déterminé. La seconde permet de restaurer ces blocs sur le disque à partir du fichier. L'habitude que j'ai prise de sauver dans un sous-catalogue avec ces deux commandes la piste 0 de mes disques de travail m'a plusieurs fois 'sauvé la vie'...

Dans toutes ces commandes, il y en a une que je cherche toujours : on sait que ProDOS ne disposera d'une commande "Format" universelle, permettant de formater tout disque ProDOS, qu'à partir de la (future...) version 2.0 du ProDOS 16. On peut donc s'étonner de voir que seul Pom's a commencé à combler ce manque par une commande INIT (Pom's N° 20) pour les disques 5'25. Mais on ne dispose toujours pas d'une commande "Format" universelle : la chose est d'autant plus étonnante qu'Apple vend aux développeurs le source d'un formateur universel. Qui va se décider à la faire ?

## DIVERSI-COPY, DIVERSI-CACHE et les autres

Le "cache" est une méthode pour accélérer les accès disques en 'cachant' dans un coin inutilisé de la mémoire des données fréquemment lues ou utilisées, pour les ressortir instantanément en évitant l'accès disque demandé. Le cache a été un des thèmes débattus entre développeurs à la fête de l'Apple // de San Francisco, et il semble bien que le IIGS inclura dans l'avenir des caches non seulement disque, mais mémoire (en gros, il est plus rapide de stocker vos fenêtres au grenier et de les redescendre quand vous en avez besoin, que de les détruire pour les reconstruire ensuite à chaque fois pièce par pièce).

Pom's 34 (page 66) a déjà recommandé aux utilisateurs de IIGS l'ensemble d'utilitaires de **Bill Basham** diffusé en 'Shareware'. **Diversi-Copy** est le copieur 3'5 le plus rapide sur tout Apple //, et **Diversi-Cache** accélère considérablement les accès disques sur IIGS avec lecteur Apple 3'5.

On sait moins que la carte mémoire **GS Ram** d'**Applied Engineering**, diffusée en France par **Bréjoux**, est accompagnée d'un utilitaire de cache-mémoire, tout comme la carte **Rampack 4 GS** d'**Orange Micro**. Et que la carte **AST SPRINTDISK**, qu'on peut trouver à des prix de braderie (voir Pom's 34 page 64), est non seulement une carte d'extension mémoire type Apple pour tous les Apple //, mais fait aussi du cache.

## Pour lire et réparer les disques

Lire les disques directement (blocs et même pistes brutes), vous devrez le faire un jour pour réparer les disques abîmés. **Prosel** et les deux **bitcoplers** (5'25 et 3'5) de **Copy // Plus** font l'essentiel des tâches d'édition des blocs. Qui veut un éditeur de blocs pratique et peu encombrant peut choisir **Diskworks**, diffusé en Shareware par **Living Legend Software**.

**Prosel** dispose en plus d'un excellent utilitaire de récupération des fichiers détruits et des catalogues effacés, qui récupère en particulier automatiquement les sous-catalogues. Si vous avez bien organisé votre disque, c'est l'arme absolue contre l'usure du bloc 2.

Un seul programme rajoute des fonctions nouvelles - et indispensables - qui ne sont pas toutes dans Prosel : il s'agit de **Bag Of Tricks //** de **Quality Software**, excellent utilitaire de récupération des disques endommagés en ProDOS, DOS 3.3, DOS 3.2 (!), CP/M et Pascal. Il se compose de quatre modules :

— **Trax** édite les pistes brutes sur les disques 5'25 et permet à ceux qui s'y connaissent de repérer les formatages bizarres et les erreurs ;
— **Init** permet de reformater individuellement une piste d'un disque 5'25 en conservant les informations lisibles, donc de récupérer l'essentiel d'un fichier quand il n'a que quelques blocs endommagés ;
— **Zap** vous permet ensuite de lire et changer les blocs et secteurs, avec d'excellentes macro-commandes qui en font un véritable langage d'édition de blocs ;
— **Fixcat** permet de récupérer les catalogues détruits en DOS et ProDOS et de ressusciter des fichiers totalement disparus du catalogue. Ce dernier programme à lui seul peut vous éviter de terribles ennuis. Un indispensable.

## Que choisir ?

Pour ne pas "noyer" le lecteur, je conclurai par une sorte de classement des programmes évoqués en deux catégories :

- les indispensables tout d'abord : Prosel, Copy // Plus, Bag of Tricks //, les utilitaires Diversi, le Bird's Better Bye, Big U, Uniram, et... la collection de Pom's ! Pour les développeurs sur IIGS, rajoutez encore APW/ORCA ;
- les utiles ensuite, ceux qui apportent un confort supplémentaire mais ne sont pas absolument indispensables si vous avez les premiers : The Desktop Manager avec le TDM Toolkit pour GS, Runrun pour //e - //c, ProCommand, Diskworks, Mousefiler.

Avec cette batterie d'utilitaires, vous serez impeccablement armé pour la gestion de vos disques ProDOS. Mais n'oubliez jamais que si tous ces bons programmes sont possibles, c'est parce qu'ils prennent appui sur l'excellence d'un autre, dont j'espère vous avoir prouvé la force : ProDOS, d'Apple Computer.

Où trouver les produits cités ici ? ...page 33 ➞

# un New Desk Accessory

# Converter

Vincent & Bernard TOMENO

## Programme Converter.Pas
### Source TML. Pascal

```
(*=================================================================================*)
(*---  CALCULATRICE HEXADECIMALE <==> BINAIRE <==> DECIMALE pour APPLE II GS   ---*)
(* New DeskAccessory par Vincent et Bernard TOMENO ( sous TML PASCAL ,fevrier 88 )*)
(*=================================================================================*)

(*$DeskAcc -1 -1 Converter*)                                    (*$LongGlobals+*)

PROGRA  Converter;
USES    QDIntf,GSIntf,MisoTools,ConsoleIO;

VAR     myWindOpen:     Boolean;
        myWind:         NewWindowParamBlk;
        myWindPtr:      WindowPtr;
        touches:        Array[0..20] of record
                              code:  Char;
                              r:     Rect;
                        end;
        affichage:      string[16];
        CurrentType :   integer;
        firstClick  :   boolean;

(*=================================================================================*)
(* ------------------- Routines specifiques du Desk Accessory ------------------ *)
(*=================================================================================*)

Procedure resultat;(* ---Routine des calculs et des conversions --- *)
var i,j,A,B,C,D: integer;
    nbr,nbraux : real;
    r2         : rect;
    affichage2 : string[16];
begin
    affichage2 := affichage;
    if CurrentType = 17 then begin                    (* on entre un chiffre binaire *)
        for i := 1 to 16-length(affichage2) do affichage2 := concat('0',affichage2);
        A := 0;
        if affichage2[1] = '1' then A := 8;
        if affichage2[2] = '1' then A := A+4;
        if affichage2[3] = '1' then A := A+2;
        if affichage2[4] = '1' then A := A+1;
        B := 0;
        if affichage2[5] = '1' then B := 8;
        if affichage2[6] = '1' then B := B+4;
        if affichage2[7] = '1' then B := B+2;
        if affichage2[8] = '1' then B := B+1;
        C := 0;
        if affichage2[9] = '1' then C := 8;
        if affichage2[10] = '1' then C := C+4;
        if affichage2[11] = '1' then C := C+2;
        if affichage2[12] = '1' then C := C+1;
        D := 0;
        if affichage2[13] = '1' then D := 8;
        if affichage2[14] = '1' then D := D+4;
        if affichage2[15] = '1' then D := D+2;
        if affichage2[16] = '1' then D := D+1;
        nbr := A*4096+B*256+C*16+D;
    end
    else if CurrentType = 16 then begin               (* on entre un chiffre HEXA *)
        for i := length(affichage2)+1 to 4 do
            affichage2 := concat('0',affichage2);
        if affichage2[1] in ['0'..'9'] then A := ord(affichage2[1])-48
        else A := ord(affichage2[1])-55;
        if affichage2[2] in ['0'..'9'] then B := ord(affichage2[2])-48
        else B := ord(affichage2[2])-55;
        if affichage2[3] in ['0'..'9'] then C := ord(affichage2[3])-48
        else C := ord(affichege2[3])-55;
        if affichage2[4] in ['0'..'9'] then D := ord(affichage2[4])-48
        else D := ord(affichage2[4])-55;
```

CONVERTER FONCTIONNE SOUS PRODOS 16 SUR LES Apple IIGS.

**Dans le menu , sur le bureau électronique (voir la photo de couverture), avec toute application GS, voici un nouvel accessoire de bureau écrit en TML Pascal.**

Une des tâches fréquentes (mais néanmoins ardue !) du programmeur est de faire des conversions :
Décimal ↔ Hexadécimal ↔ Binaire

CONVERTER est un nouvel accessoire de bureau pour le menu  de l'Apple IIGS – écrit en TML Pascal – qui réalise ces opérations. Il se présente comme une calculette avec des touches à 'cliquer' (0 → F) ; les résultats (binaires, décimaux et hexadécimaux) s'affichent dans trois cases différentes ; la dernière case est constituée d'un pavé coloré qui vous donne (modulo 16) la teinte correspondante au chiffre entré (pratique pour se rappeler que 0 correspond au noir, 4 au vert foncé, etc.).

Converter est sur la disquette Pom's 35 (avec son source Converter.Pas) : il suffit de le transférer dans le directory DESK.ACCS d'un Système ProDOS 16 pour le voir à l'œuvre .

Converter est donc écrit en TML Pascal, qui est un Éditeur-Compilateur sur disquette 3'5 ; il coûte $125 et est puissant, rapide en compilation, beaucoup plus facile à manier que le GSBasic ou le redoutable APW : tout est en gestion-souris et menus déroulants. La syntaxe est celle du Pascal conventionnel avec des instructions nouvelles permettant de gérer les outils du IIGS et programmer assez facilement l'écran SHGR, les barres de

menus, les fenêtres, les graphiques, etc.

Ce Pascal n'a de 'Pascalien' que le nom, la syntaxe et la structure des sources : il est en effet sur une disquette au format ProDOS et génère des programmes ProDOS de type $B3 ($B8 dans le cas particulier des 'new desk accessories') . Il est accompagné de nombreux exemples de sources dont on peut largement s'inspirer : c'est ce que nous avons fait pour écrire CONVERTER à partir d'autres accessoires de bureau présents sur le disque (TMLClock, TMLCalculator, TMLPuzzle). Faire du Couper/Coller entre plusieurs 'sources' affichés ensemble à l'écran est bien entendu possible .

Un inconvénient : les programmes-objets sont assez volumineux (bien plus de blocs que le source) .

## Quelques mots sur le source Converter.Pas

Le Préambule – déclaration des "USES" et "VAR" globales – est classique.

Trois procédures sont propres à CONVERTER :

- procédure *résultats* où l'on fait tous les calculs de conversions (peut-être aurait-on pu raccourcir cette partie du source en faisant appel aux outils mathématiques du GS "SANE" et "INTEGER") ;
- procédure *DoMouseDown* : gestion des événements souris ;
- procédure *DrawPanel* : on dessine les éléments de la calculette.

Quatre procédures sont obligatoires et donc communes à tous les accessoires de bureau : DaInit, DaClose, DaOpen, DaAction (elles comportent aussi quelques instructions propres à Converter).

**TML Systems**
4241 Baymeadows Rd., Suite 23,
Jacksonville, Florida 32217, USA

```
        nbr := A;
        nbr := nbr*4096+B*256+C*16+D;
    end
    else if CurrentType = 18 then begin                (* on entre un chiffre DECI *)
        nbr := 0;
        for i := 1 to length(affichage2) do nbr := nbr*10+ord(affichage2[i])-48;
        A := TRUNC(nbr / 4096);
        B := TRUNC(nbr / 256) MOD 16;
        C := TRUNC(nbr / 16) MOD 16;
        D := TRUNC(nbr) MOD 16;
    end;
    for i := 16 to 18 do with touches[i] do begin (* on fait les conversions *)
        if i=16 then begin                                        (* affichage hexa *)
              if A<10 then affichage2 := concat(chr(A+48),'')
              else affichage2 := concat(chr(A+55),'');
              if B<10 then affichage2 := concat(affichage2,chr(B+48))
              else affichage2 := concat(affichage2,chr(B+55));
              if C<10 then affichage2 := concat(affichage2,chr(C+48))
              else affichage2 := concat(affichage2,chr(C+55));
              if D<10 then affichage2 := concat(affichage2,chr(D+48))
              else affichage2 := concat(affichage2,chr(D+55));
        end
        else if i=17 then begin                                   (* affichage binaire *)
              affichage2 := '';
              nbraux := nbr;
              while nbraux <> 0 do begin
                        affichage2 := concat(chr((TRUNC(nbraux) MOD 2)+48),affichage2);
                        nbraux := TRUNC(nbraux / 2);
              end;
              for j := length(affichage2)+1 to 16 do affichage2 := concat('0',affichage2);
        end
        else if i=18 then begin                                   (* affichage decimal *)
              nbraux := nbr;
              if nbraux = 0 then affichage2 := '0'
              else affichage2 := '';
              while nbraux <> 0 do begin
                        affichage2 := concat(chr((TRUNC(nbraux) MOD 10)+48),affichage2);
                        nbraux := TRUNC(nbraux / 10);
              end;
        end;
        SetDithColor(15);                                   (*     on     *)
        SetRect(r2,r.left,r.top,r.right,r.bottom);          (*            *)
        PaintRect(r2);                                      (*     va     *)
        MoveTo(r.left+2,r.bottom-3);                        (*            *)
        DrawChar(touches[i].code);                          (* dessiner   *)
        for j := 1 to length(affichage2) do begin           (*            *)
              MoveTo(r.right-j*9-3,r.bottom-3);             (*    les     *)
              DrawChar(affichage2[length(affichage2)+1-j]);  (*            *)
        end;                                                (* resultats  *)
    end;
    with touches[19] do begin          (* la couleur du pave = resultat MOD 16 *)
        SetDithColor((TRUNC(nbr)) MOD 16);              (* c'est pratique pour trouver *)
        PaintRect(r);                                   (* le numero d'une couleur !   *)
        SetDithColor(15);
    end;
    firstClick := true;
    affichage := '';
end; (* of Resultat *)

Procedure DrawPanel(init : boolean); (* dessine et redessine la calculette *)
var     i,j,debut,fin:  Integer;
        r2:             Rect;
begin
    if  init then begin
        SetDithColor(6);
        SetRect(r2,0,0,176,89);
        PaintRect(r2);
        debut := 0;
        fin := 20;
    end
    else begin
        debut := 16;    fin := 18;
    end;
    For i := debut to fin do with touches[i].r do begin
        if    i <> 19 then begin
              SetDithColor(0);
              SetRect(r2,left+2,top-1,right+3,bottom-1);
              PaintRect(r2);
        end;
        if (i=17) and Init then SetDithColor(15)
        else if (i in [16..18]) and (i <> CurrentType) then SetDithColor(3)
        else if init and (i = 19) then SetDithColor(0)
        else SetDithColor(15);
        SetRect(r2,left,top,right,bottom);
        PaintRect(r2);
        if i < 19 then begin
              MoveTo(left+5,bottom-3);
              if (i=17) and init then drawstring(
                        concat('     ',chr(17),' TomSoftWare ',chr(17)))
```

CE PROGRAMME PASCAL (12089 OCTETS) GENERE UN ACCESSOIRE DE 15872 OCTETS HELAS IMPOSSIBLE A LISTER DANS LA REVUE. RENDEZ-VOUS SUR LA DISQUETTE POM'S 35.

```
            else DrawChar(touches[i].code);
        end
        else if i = 20 then begin
            MoveTo(left+17,bottom-3);                DrawString('C O N V E R T');
        end
    end;
end; (* of DrawPanel *)

Procedure DoMouseDown(theEvent: EventRecord);          (* on a clique sur la souris !!! *)
var i,j;    Integer;    chknum: Integer;               r:      Rect;
begin
    GlobalToLocal(theEvent.where);
    if firstClick then with touches[CurrentType] do begin
        PaintRect(r);
        MoveTo(r.left+5,r.bottom-3);
        DrawChar(code);
        firstclick := false;
        DrawPenel(false);
    end;
    for i := 0 to 15 do with touches[i] do
        if PtInRect(theEvent.where,r) then
            if (  (  (CurrentType = 16) and (length(affichage)<4) )
                or( (CurrentType = 17) and (length(affichage)<16) and
                    ( (i=0) or (i=1) ) )
                or( (CurrentType = 18) and (i in [0..9])
                    and(  (length(affichage)<9) or
                        ( (length(affichage)=4) and
                            (  (affichage<'6553')
                            or(    (affichage='6553')
                                and (i in [0..5]) ) ) ) ) ) )
        then affichage := concat(affichage,code);
    for i := 16 to 18 do with touches[i] do
        if    PtInRect(theEvent.where,r) then begin
              SetDithColor(15);
              PaintRect(r);
              MoveTo(r.left+5,r.bottom-3);
              DrawChar(code);
              affichage := '';
              CurrentType := 1;
              DrawPanel(false);
    end;
    for i := 1 to length(affichage) do begin
        MoveTo(touches[CurrentType].r.right-i*9-3,touches[CurrentType].r.bottom-3);
        DrawChar(affichage[length(affichage)+1-i]);
    end;
    if PtInRect(theEvent.where,touches[20].r) then resultat;

end; (* of DoMouseDown *)


(*-----------------------------------------------------------------------------*)
(* ---------- Les 4 routines indispensables pour tout Desk Accessory  ---------- *)
(*-----------------------------------------------------------------------------*)

Function DAOpen: WindowPtr;
begin
    if myWindOpen then
        SelectWindow(myWindPtr)
    else begin
        with myWind do begin
              param_length := sizeof(NewWindowParamBlk);
              wFrame       := $C0A0; (* type de la window et ses proprietes *)
              wTitle       := à'Hex<->Bin<=>Dec';
              wPosition.top    := 40;  wPosition.bottom := 129;
              wPosition.left   := 20;  wPosition.right  := 196;
              end;
        myWindPtr := NewWindow(myWind);
        SetSysWindow(myWindPtr);
        end;

    DAOpen      := myWindPtr;
    myWindOpen := true;
end; (* of DAOpen *)

Procedure DAClose;
begin
    CloseWindow(myWindPtr);    myWindOpen := false;
    firstClick := true;    CurrentType := 17;              affichage := '';
end; (* of DAClose *)

Procedure DAAction(Code: Integer; Param: Longint);
var currPort: GrafPtr;what: Integer;   modifiers: Integer;
begin
    case Code of
        DAEvent: begin
              currPort := GetPort;
              SetPort(GrafPtr(myWindPtr));
              what := EventRecordPtr(param)^.what;
              if what = updateEvt then begin
```

➡ Suite de la page 30

**Apple Coop**
290 S.W. 43rd Street, Renton, WA 98055, USA

**Applied engineering**
P.O. Box 798, Carrolhon, TX 75006, USA

**Beagle Bros - Software Touch**
6215 Ferris Square, Suite 100, San Diego, CA 92121, USA

**Bréjoux**
29, rue Montribloud – 69009 Lyon – ☎ 7836 5269

**Byte Works Inc.**
4700 Irving Boulevard NW, Suite 207, Albuquerque, NM 87114, USA

**Central Point Software**
9700 S.W. Capitol Highway, £100, Portland, OR 97219, USA

**Checkmate Technology**
509 South Rockford Drive, Tempe, AZ 85281, USA

**Diversified Software Research**
34880 Bunker Hill, Farmington, MI 48018-2728, USA

**Froggy Software**
33, Avenue Philippe Auguste – 75011 Paris ☎ 43 58 25 98.

**Klan Software Inc.**
1850 Union Street £183, San Francisco, CA 94123

**Living Legend Software**
1915 Froude Street, San Diego, CA 92107, USA

**Nibble Publications**
52 Domino Drive, Concord, MA 01742-9906, USA

**On Three**
4478 Market Street, Suites 701-702, Ventura, CA 93003, USA

**Orange Micro**
1400 N. Lakeview Avenue, Anaheim, CA 92807, USA

**Pinpoint Publishing**
5901 Christie Avenue, Emeryville, CA 94608, USA

**Prim'vert/DDA**
36, rue des États Généraux – 78000 Versailles

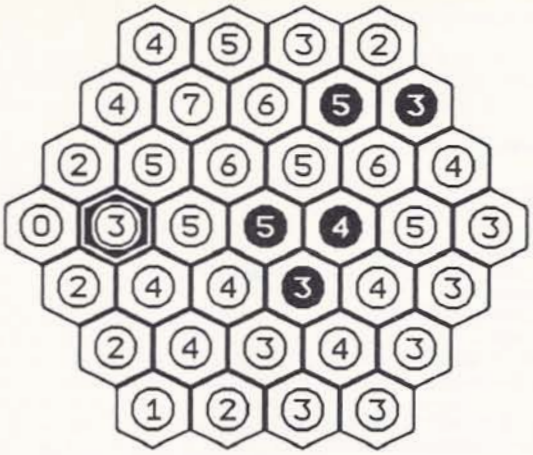
**Quality Software**
21610 Lassen Street, £7, Chatsworth, CA 91311, USA

**Roger Wagner Publishing**
1050 Pioneer Way, Suite P, El Cajon, CA 92020, USA

**Star Development Co.**
P.O. Box 5067 Elm Grove, WI 53122, USA

**Steven Humpage**
2427 NE 24th Avenue, Portland, OR 97212, USA

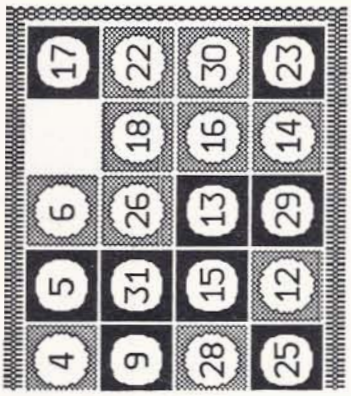
```
                BeginUpdate(myWindPtr);
                DrawPanel(true);
                EndUpdate(myWindPtr);
                end
            else if what = mouseDown then begin
                DoMouseDown(EventRecordPtr(param)^);
                end;
            SetPort(currPort);
            end;
        DARun: begin
            end;
        DACursor: ;
        DAMenu: begin
            end;
        DAUndo,
        DACut,
        DACopy,
        DAPaste,
        DAClear:  Code := 1;
    end;
end; (* of DAAction *)

Procedure DAInit(Code: Integer);
var lar,haut,Hint,Vint,i,j: integer;
begin
    lar := 20;haut := 13;Hint := 4;Vint := 4;
    if Code = 0 then begin
        (* A DeskShutDown Call, chk that my window is closed *)
        if myWindOpen then DAClose;
        end
    else begin
        (* A DeskStartUp Call, init myWindOpen flag *)
        myWindOpen      := false;
        (* Definissons les touches de la calculette ...*)
        for i := 1 to 2 do for j := 0 to 6 do
            SetRect(touches[(i-1)*7+j].r,5+j*(lar+Hint),(i+2)*(haut+Vint)-haut,
                        5+j*(lar+Hint)+lar,(i+2)*(haut+Vint));
        for i := 0 to 1 do
            SetRect(touches[i+14].r,5+i*(lar+Hint),5*(haut+Vint)-haut,
                    5+i*(lar+Hint)+lar,5*(haut+Vint));
        for i := 0 to 9 do touches[i].code := chr(i+48);    (* touches 0 a 9 *)
        for i := 1 to 6 do touches[i+9].code := chr(i+64);  (* touches A a F *)

        SetRect(touches[16].r,15+4*(lar+Hint),haut+Vint*2,
                    5+6*(lar+Hint)+lar,2*(Vint+haut));
        touches[16].code := '$';                            (* touche hexadecl *)

        SetRect(touches[17].r,5,Vint,
                    5+6*(lar+Hint)+lar,Vint+haut);
        touches[17].code := '%';                            (* touche binaire *)

        SetRect(touches[18].r,5,haut+Vint*2,
                    4+2*(lar+Hint)+lar,2*(Vint+haut));
        touches[18].code := 'd';                            (* touche decimale *)

        SetRect(touches[19].r,8+3*(lar+Hint),haut+Vint*2,
                    13+3*(lar+Hint)+lar,2*(Vint+haut)-1);
        touches[19].code := ' ';                            (* touche coloree *)

        SetRect(touches[20].r,5+2*(lar+Hint),5*(haut+Vint)-haut,
                    5+6*(lar+Hint)+lar,5*(haut+Vint));
        touches[20].code := ' ';                            (* touche Convert *)

        affichage := '';CurrentType := 17;firstClick := true;
        end;
end; (* of DAInit *)

(*=================================================================================*)
(* ---------------------------------  MAIN  ---------------------------------------*)
(*=================================================================================*)
BEGIN
 (* PAS DE PRGM PRINCIPAL... les procedures d'un NDA sont appelees automatiquement *)
END.
(*=================================================================================*)
(*=================================  TOMSOFTWARE FEVRIER 1988  ==================*)
(*=================================================================================*)
```



# Ludologic

Taquin, Noir & blanc, hexagone magique, trois jeux de réflexion de difficulté croissante. Ces jeux qui nécessitent des neurones aussi calmes qu'entraînés, ne devraient pas décevoir les amateurs de puzzles et autres casse-têtes.

*Pour les Apple //*
*80,00 F franco,*
*Bon de commande page 71*
*(La disquette comprend les sources)*




**(1) 39 53 04 40** CE SERVEUR MINITEL — GRATUIT — EST UN LIEN ENTRE DEUX NUMEROS DE POM'S. IL VOUS PERMET DE TESTER LE "REPOM'DEUR" DE POM'S34, DE LAISSER DES MESSAGES POUR LA REDACTION ET VOUS INFORME DES NOUVEAUTES...

# Toujours plus vite...

# Compilateur Beagle v2.2

**Difficile de présenter le compilateur Beagle sans tomber dans l'excès de superlatifs (pourtant mérités !). Le compilateur Beagle est en effet le premier à fonctionner sous ProDOS et le premier à générer un code moins volumineux que le Basic ; de plus il offre la 'compilation instantanée' qui allie la souplesse du Basic aux avantages du compilateur.**

## Le but du compilateur

Il s'agit de transformer un programme écrit en Basic Applesoft en un code machine : l'avantage immédiat étant une exécution considérablement accélérée. En fait, Beagle a opté pour la solution du *RunTime* (disons : interpréteur) : un fichier de 6Ko doit être présent en mémoire pour permettre l'exécution des programmes compilés. Les programmes Basic ne sont donc pas transformé en code machine 6502 directement exécutable mais en un *langage* que le RunTime exécutera : tout ceci est transparent pour l'utilisateur.

Outre la vitesse d'exécution, deux avantages sont à noter : le code est plus concis que le Basic original donc il laisse plus de place aux variables lorsqu'il est en mémoire, et plus de place sur le disque. Il offre également une protection contre les indélicats qui voudraient s'inspirer d'un algorithme utilisé dans le programme par exemple, car un fichier compilé est tout-à-fait hermétique, bien plus qu'un programme assembleur. Pour en comprendre le fonctionnement sans le 'source' Basic, il faudrait avant tout analyser le fonctionnement du RunTime...

## L'utilisation

Alan Bird, l'auteur, a simplifié la tâche de l'utilisateur : il n'est pas nécessaire de jongler avec les disquettes pour éditer puis compiler puis essayer puis rééditer...

Il suffit de démarrer sur la disquette Beagle pour installer en mémoire le compilateur et l'interpréteur (la documentation indique clairement comment faire pour se dispenser de démarrer sur cette disquette).

Maintenant pour exécuter un programme Basic, il suffit de faire :

`RUN programme` ou

`- programme`

Le programme se charge en mémoire, se compile en un instant et s'exécute. Pour modifier le programme, faire :

`LOAD programme`

l'éditer comme d'habitude puis le sauvegarder par :

`SAVE programme`

enfin le réessayer avec l'ordre `RUN programme`.

Lorsque le programme est au point, on peut le sauvegarder sur disque en format compilé (ce qui supprime la courte phase de compilation) ; pour lancer ce programme déjà compilé, il suffit de taper :

`RUN programmecompilé`

## La compilation

Le temps de compilation est particulièrement bref : 7 à 8 secondes pour un programme de 10Ko ce qui correspond déjà un programme Basic volumineux. Dans la majorité des cas la compilation dure moins de deux secondes.

Les fautes de syntaxe, les GOTOs fantaisistes, les erreurs dans les dimensionnements de tableaux sont détectés lors de la compilation et l'erreur est indiquée clairement à l'écran. Il est bien sûr possible de passer outre ce contrôle au cas où l'erreur supposée était volontaire ; mais bien souvent, le compilateur signale des erreurs qui étaient passées inaperçues malgré des tests répétés.

Lorsque des données ou des routines en langage machine sont discrètement cachées à la fin du programme Basic, le compilateur propose de l'inclure au code compilé. Parfois, il ne s'agit que de 'garbage' inutile et l'on peut demander sa suppression.

## Toutes les instructions

Le compilateur ne s'émeut pas des instructions HIMEM: LOMEM: et autres DIM A$(variable). Même les instructions ampersand — sans paramètre — sont acceptées directement ; seules les instructions typiquement liées aux lignes Basic (LIST 10-20, TRACE ou DEL 100 par exemple) et aux entrées/sorties sur cassettes sont incompatibles mais cela ne devrait pas vraiment déranger !

Pour les programmes unis par l'ordre CHAIN ou ceux qui utilisent STORE ou RESTORE, la compilation ne pose également pas de problèmes ce qui est encore une belle performance.

Les routines ampersand avec paramètres sont acceptées par le compilateur mais le source de la routine doit être modifié pour évaluer les paramètres comme indiqué par la documentation. C'est plus complexe, mais Alan Bird l'a prévu.

Bien souvent, avec ProDOS, on loge les routines assembleur à la place habituelle du Basic (à partir de l'adresse $801 – voir Pom_Link dans le n° 34) et on déplace le programme Basic à une adresse supérieure ($2000 par exemple). Cette façon de faire est également prise en compte avec le compilateur Beagle avec l'ordre :

`RUN programme, A$2000` par exemple.

Il existe un type de programme dont on est sûr qu'il n'est pas compilable : celui qui s'auto-modifie. C'est le cas du programme PARAMETRIQUE du numéro 24 de Pom's où des fonctions sont 'pokées' dans une ligne de remarque.

## Les performances

La première performance de ce compilateur est d'exécuter sans modification les programmes Basic :

- booter sur le compilateur,
- prendre la disquette Pom's 29,
- faire `RUN SIMPLEXE` et on se trouve avec un Apple turbo-compressé : les résolutions sont quasi-instantanées.

En ce qui concerne les 'chronos', voici quelques mesures effectuées sur un Apple // (sur un IIGS, diviser les temps par 2,8) :

### L'exemple frappant par excellence

```
10 FOR I = 1 TO 10000:U = PEEK (49200):
   NEXT
```

*Basic : 74 s.*
*Compilé : 3,5 s.*

### Pom's 25, Tri de chaînes

Nombre de reclassement en une minute du tableau exemple de 10 x 3 (pays, capitales, abréviations) :

*Basic : 28*
*Compilé : 84*

Plus étonnant, le programme Basic avec routine de tri écrite directement en assembleur ne donne que 60 ! Affaire à suivre...

### Programmes Beagle

L'un des programmes de démonstration sur la disquette Beagle : il s'agit de la simulation du jet de deux dés avec statistiques sur la fréquence d'apparition du total 7.

*Basic : 302 essais/mn*
*Compilé : 870 essais/mn*

La différence est bien moins frappante lorsqu'il y a de nombreux accès à la ROM de l'Apple : compilé ou non, le calcul de SIN (x) prendra le même temps. Ce phénomène apparaît avec l'un des programme de démonstration qui utilise les fonctions XDRAW et SCALE intensivement (rotation d'un trotteuse à l'écran).

*Basic : 18 tours/mn*
*Compilé : 35 tours/mn*

### Pom's 25, Tesseract

Ce programme de dessin en haute résolution d'objet en quatre dimensions sollicite également beaucoup la ROM : le gain après compilation reste honorable :

*Basic : 60 dessins/mn*
*Compilé : 94 dessins/mn*

### Pom's 31, Le voyageur

C'est la résolution du problème classique du voyageur de commerce qui optimise sa tournée (ici entre cinq villes).

*Basic : résolution en 24 s.*
*Compilé : résolution en 3,5 s.*

Sur la disquette, le Basic occupe 19 blocs et le fichier compilé 12...

Temps de compilation : 8 s.

## Des 'goodies'

Trois utilitaires sont livrés sur la disquette Beagle : Slow.Paddles, Input.Anything et Fast.Hplot.

- Le premier permet de ne pas lire trop rapidement la valeur des poignées de jeux pour en garder la cohérence.
- Input.Anything remplace l'INPUT du BASIC par une routine qui accepte virgules et deux-points : pratique. Cette routine occupe 256 octets.
- Fast.Hplot optimise la fonction HPLOT au prix de 1280 octets supplémentaires en mémoire. Voici les temps relevés pour remplir la page HGR avec le programme suivant :

```
10 HGR2 : HCOLOR= 7
20 FOR X = 0 TO 279
30 FOR Y = 0 TO 191
40 HPLOT X,Y
```

*50 NEXT : NEXT*
*Basic : 197 s.*
*Compilé : 36 s.*
*Compilé + Fast.Hplot : 25 s.*

## De la place

Le compilateur et l'interpréteur chargés, 24 060 octets sont disponibles pour le programme mais... si l'ordinateur est un 128Ko ce chiffre passe à 47 162. Toute la mémoire auxiliaire n'est pas utilisée mais cela devient déjà très confortable.

L'idéal est d'avoir une extension mémoire de type Apple ou RamFactor : le compilateur Beagle les reconnaît et y stockera quantité de variables. A noter que l'utilisation de mémoires auxiliaires reste facultative (le traitement des chaînes en est très légèrement moins rapides) et que les cartes MultiRam ou RamWorks peuvent être partitionnées par l'utilisateur pour en affecter une partie au disque virtuel et une partie au compilateur.

## Faire plus

Trouver le point d'entrée d'une routine Applesoft n'est pas bien simple, la ROM étant truffée de *jump* et de ruses en tout genre. À l'aide du compilateur (et surtout de sa documentation qui donne l'ensemble des points d'entrée), il est possible de détourner l'interprétation des commandes Applesoft à son profit sans jouer avec CHRGET et, par exemple, de doter l'Applesoft de la fonction LPRINT. La notice montre comment transformer le HOME pour qu'il agisse comme HGR...
La notice donne par ailleurs des informations sur le stockage des variables pour en permettre l'utilisation depuis une routine personnelle.

## Présentation, documentation

Pom's distribue le compilateur sous la forme suivante :

- le *package* américain comprenant une disquette 140Ko non protégée et la documentation américaine ;
- la documentation française signée Pom's (traduction *in extenso* et non pas simplement un résumé). Insistons sur la clarté de la documentation qui ne requiert aucune connaissance technique ;
- et... si nécessaire, l'habituel coup de main téléphonique. Un *plus* de Pom's.

---

*Utilitaires*

# Big U de Beagle

*Damien Nould*

La disquette Big U (U-tilitaires) de Randy Brandt — l'auteur de SuperMacroWorks, c'est tout dire — est un ensemble de programmes et de commandes externes ProDOS. Elle est destinée aux programmeurs en Basic pour leur simplifier la tâche dans le travail sous ProDOS. Cette disquette regroupe des utilitaires qui utilisent pleinement les possibilités de ProDOS, mais de ProDOS 8. Les possesseurs de GS ne trouveront rien de spécifique à leur machine : Big U leur servira lors de la conception de programmes Applesoft avec Basic.System.

Avant de passer à la description sommaire des possibilités de Big U, on peut noter quelques problèmes bien connus des programmeurs auxquels Randy Brandt apporte ses solutions.

Peut-être la plus douloureuse des faiblesses de l'Applesoft car elle affecte l'interactivité des programmes : l'Input. Maintenant les caractères de contrôle sont éliminés, la touche Del fonctionne, la longueur de saisie est maîtrisée (tant en 40 qu'en 80 colonnes).

Autre faiblesse : comment créer des écrans affichables simplement depuis le Basic ? Lorsqu'il faut afficher un écran de présentation, une page d'accueil, un écran d'aide, charger la page complète depuis le disque (qui peut être le disque /RAM, donc instantané) est bien plus simple que se battre avec des PRINTs surtout lorsqu'on attache une importance à la présentation. De surcroît, encombrer le Basic avec quantité de PRINTs est un inconvénient supplémentaire. Big U offre un programme d'édition complet et, pour charger les écrans depuis un programme, il suffit de faire `-mon.ecran`.

Insuffisance non de l'Applesoft, mais d'Apple : peut-être utilisez- vous le programme Filer comme utilitaire de copie. Si c'est le cas, le programme File.Mover de Big U à lui seul vaut le prix de la disquette...

Big U use beaucoup de la possibilité d'adjoindre des commandes externes au Basic.System : ces commandes, toujours sous la main, utilisables directement au clavier ou par programme : Certaines ont déjà fait l'objet d'articles dans Pom's (Online, Copy), d'autres sont originales (Anycat, See, Show, Xlister).

# De nouvelles commandes externes ProDOS

**Anycat** pour lister le catalogue tant des disquettes DOS 3.3 que ProDOS.

**Copy** pour copier des fichiers d'un disque à un autre (une autre version est livrée en cas d'utilisation avec un seul lecteur).

**Dump** pour recopier l'écran 40 et 80 colonnes sur l'imprimante. La commande elle-même détermine quel est le format de l'écran.

**EST** est un ensemble de trois commandes ProDOS :

- **Erase** efface le programme Applesoft, réinitialise les pointeurs du Basic et le plan d'occupation mémoire de ProDOS, désactive la carte 80 colonnes, etc. Cela peut être comparé à la commande FP sous DOS 3.3.
- **Space** affiche la place libre sur le disque.
- **Time** donne date et heure.

**Hex** pour installer quatre nouvelles commandes. La première exécute les conversions entre décimal, binaire et hexa. La deuxième est une version améliorée de la commande L du moniteur (on peut préciser le nombre de lignes de désassemblage, que les lignes comportent 1, 2 ou 3 octets). La troisième affiche les registres du 6502 et le registre d'état. La dernière permettant d'afficher une zone mémoire.

**Mon** et **Nomon**, l'équivalent du Mon utilisé sous DOS 3.3 pour contrôler les opérations-disques. Les commandes Create, Open, Read etc. s'affichent à l'écran pour permettre la surveillance de leur exécution et donc... de trouver le pourquoi de la panne.

**Online**, une commande classique pour lister les volumes reconnus par ProDOS.

**Renew** pour retrouver les programmes intempestivement effacés par New.

**See** pour lister des programmes Basic sur disquette sans devoir les charger en mémoire. En fait, tous les fichiers peuvent être visualisés ainsi mais seuls les fichiers de type TXT et BAS ont bonne allure (cela correspond à la commande TYPE des IBMistes).

**Show** pour afficher directement une image du disque. Une version double-HGR est également livrée. Selon les paramètres, il est possible par exemple d'afficher la page HGR sans l'effacer, d'afficher une image HGR du disque en mode mixte texte/graphique etc.

**Xlister** met en forme les listings Basic pour les rendre plus li- sibles :

```
-----
10    D$=chr$(4)
      :HP=49200
      :UA=768
-----
20    FOR I=0 TO 10
      :  READ A
      :  POKE I,A
      NEXT
-----
```

Cette commande fonctionne à l'écran ou sur imprimante et, dans ce dernier cas, peut être adaptée à une imprimante donnée pour qu'elle passe en caractères gras compressés italiques par exemple.

# Des utilitaires

**Appender** est conçu pour concaténer des programmes Basic.

**BeeperWorks** sera utile à ceux qui sont souvent rappelés à l'ordre par AppleWorks : il en modifie le "bip" (en fréquence et durée). Uniquement pour les versions 1.1, 1.2 et 1.3.

**Bigliner** crée des lignes Basic dont le numéro est supérieur à 63999. Pratique pour y installer un message tenace.

**Cat.date** adjoint la date aux commandes Cat et Catalog.

**Cat.fixer** est un utilitaire de configuration des commandes Cat et Catalog : multi-colonnes, passage automatique en 80 colonnes.

**Cat.stepper** installe la pause indispensable pour faire le Catalog fichier par fichier ou écran par écran.

**CRT.Writer** est l'utilitaire de création d'écrans. Il comporte les fonctions nécessaires de justification, d'écriture verticale etc. y compris avec les caractères souris.

**Date.Set** met à jour la date en l'absence de carte horloge.

**Dump.40 & Dump.80** copient respectivement les écrans 40 & 80 colonnes sur l'imprimante.

**Error.Editor** est conçu pour éditer les messages d'erreurs de ProDOS (le faire sans ce programme est délicat car le codage adopté ne simplifie pas l'intervention manuelle - voir Pom's n° 28).

**File.Mover** simplifie les opérations de copie, d'effacement de fichiers et de formatage de disquettes (on ne jongle pas avec les chemins d'accès). Il comprend bien entendu toutes les commandes standard d'un utilitaire de gestion de fichiers mais également une fonction plaisante à l'usage : Move. Il s'agit de copier un fichier en effaçant l'original, donc de le déplacer : enfin une commande pour sortir simplement un fichier d'un sous-catalogue... File .Mover fonctionne également si vous utilisez vos disquettes 140Ko avec plus de 35 pistes (36...40).

**Input.40 & Input.80** relèvent le niveau du pauvre Input de l'Applesoft...

**Keycat.80** est un programme de menu accessible par le Reset : il affiche la liste des fichiers de la disquette et permet d'en charger ou d'en exécuter un par la pression d'une seule touche.

**Ram.Load** charge dans le volume /RAM des fichiers à raison de 5Ko/s sur un //e. Livré avec son complément indispensable **Ram .Save**.

**Rem.Ove** enlève les Rems d'un programme Applesoft.

**Run.Counter** contrôle le nombre d'exécutions d'un programme et en garde la trace.

**Save.40 & Save.80** sauvegardent les écrans Text sur disque.

**Super.Poke** simplifie les Pokes : pour remplir la page HGR avec $9 par exemple, faire : &POKE 8192:16384,9.

# Configuration

La plupart des programmes de la disquette Big U requièrent :

- un Apple //e avec carte 80 colonnes
- ou un Apple //c ou
- ou un Apple IIGS.

Les programmes ou commandes qui travaillent sur le disque virtuel /RAM exigent pour les //e une carte 80 colonnes étendue (c'est-à-dire un //e 128Ko).

# Présentation

Pom's vous offre non seulement le programme américain, mais également la documentation française intégrale (réalisée par Pom's) et le coup de pouce téléphonique en cas de besoin.

Prix de l'ensemble : 450,00 TTC (avec carte Joker : 400,00 TTC) pour 18 commandes externes et 24 programmes...

# Un accessoire de bureau

# Corner light

LightInstall

Alain Bohec

"Corner light" est un accessoire de bureau alors que "LightInstall" est une application, mais ces deux logiciels sont associés l'un à l'autre.

## Corner light

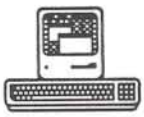
On ne sait pas toujours ce que le Macintosh est en train de faire ; il nous arrive même, lors d'accès répétés et/ou prolongés au disque dur par exemple, de croire qu'il est tout simplement 'planté'.

"Corner light" permet d'installer une routine dans le *vertical retrace manager*. Cette routine demeure en permanence dans la zone de mémoire du système. Notamment, elle ne disparaît pas lors d'un changement d'application. La présence de la routine est signalée par un point au-dessus de la ⌘.

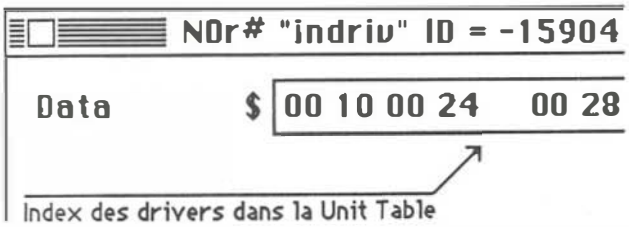
La fonction de cette routine est de faire clignoter le coin supérieur gauche de l'écran lorsque certains 'drivers' sont en service. Dans ce cas, les fonctionnalités de l'application en cours sont parfois interrompues, et l'utilisateur en est donc prévenu par "Corner light". Bien entendu, il paraît inutile de s'intéresser aux 'drivers' du type 'accessoire de bureau', d'autant moins que leurs index dans la *UnitTable* peuvent changer lors des transferts à l'aide du Font/DA Mover. Aussi, nous ne considérerons ci-dessous que les 'vrais' drivers, c'est-à-dire ceux qui ne sont pas des accessoires.

Les index des drivers examinés par "Corner light" figurent dans la ressource 'NDr#' de "Corner light" (voir Inside Macintosh volume II, chap. 6 : "The Deviee Manager"). La liste des index des drivers, telle qu'elle se trouve dans l'édition de Novembre 85 d'"Inside Macintosh", est, en hexadécimal, la suivante :

| | |
|---|---|
| $04 | Hard Disk Driver (XL seulement) |
| $08 | Printer Driver |
| $0C | Sound Driver |
| $10 | Disk Driver |
| $14 | Serial Driver Port A Input |
| $18 | Serial Driver Port A Output |
| $1C | Serial Driver Port B Input |
| $20 | Serial Driver Port B Output |
| $24 | AppleTalk .MPP Driver |
| $28 | AppleTalk .ATP Driver |
| $80 | SCSI driver 0 |
| $84 | SCSI driver 1 |
| $88 | ... |
| $9C | SCSI driver 7 |



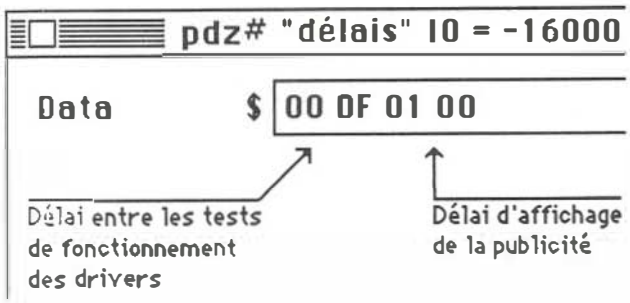
Il est possible de configurer l'accessoire à l'aide d'un éditeur de ressources :




Dans la configuration ci dessus, "Corner light" examine le *Disk Driver* et les drivers *d'AppleTalk* (indications des drivers sur 16 bits).

La routine peut être ôtée du *vertical retrace manager* en sélectionnant de nouveau "Corner light" dans le menu ⌘.

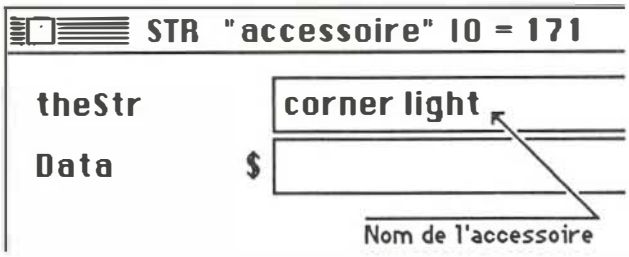
La fréquence de la routine et donc du clignotement est écrite dans les deux premiers octets de la ressource de type 'pdz#'. Les deux octets suivants représentent la durée d'apparition de l'affiche :




## LightInstall

La fonction principale de "LightInstall" est d'ouvrir l'accessoire de bureau "Corner light", ce qui a pour effet d'installer la routine de clignotement. Cette application a été conçue pour être lancée au démarrage du système.

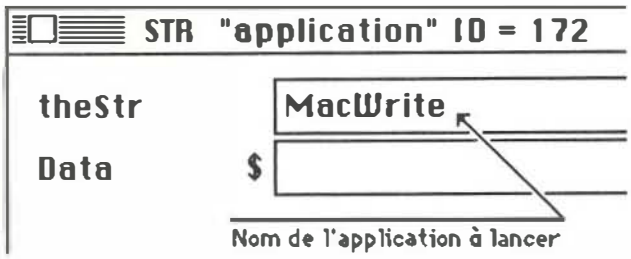
Si elle trouve l'accessoire à ouvrir (qu'elle repère par son nom interne DDO), elle l'ouvre en remplaçant éventuellement son nom par celui qui figure dans la ressource de type 'STR ' et d'ID 171.



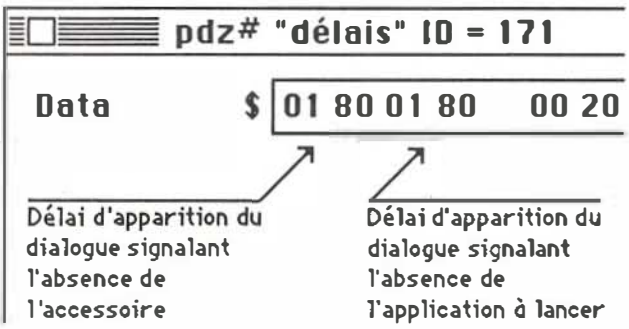

Pour que le nom apparaisse à son rang alphabétique normal dans le menu , il faut qu'il commence par le caractère nul 00. Pour insérer celui-ci, on peut ouvrir la forme hexadécimale de la ressource ci-dessus en pressant la touche 'option' pendant le double-clic.

Si elle ne le trouve pas l'accessoire à ouvrir, elle prévient par un dialogue. La durée d'affichage de ce dialogue figure dans la ressource de type 'pdz#' et d'ID 171 (voir plus bas). On passe ensuite à l'étape décrite ci-dessous :

Après avoir cherché à ouvrir l'accessoire, "LightInstall" cherche à lancer l'application dont le nom figure dans sa ressource de type 'STR' et d'ID 172.




Si elle ne trouve pas cette application, en particulier si, sur un disque formaté HFS, elle ne se trouve pas dans le même dossier, "LightInstall" prévient par un dialogue. En l'absence d'action de l'utilisateur, ce dialogue disparaît au terme d'un délai dont la valeur se trouve dans la ressource de type 'pdz#' et d'ID 171 ci-dessous, et le Finder est lancé.



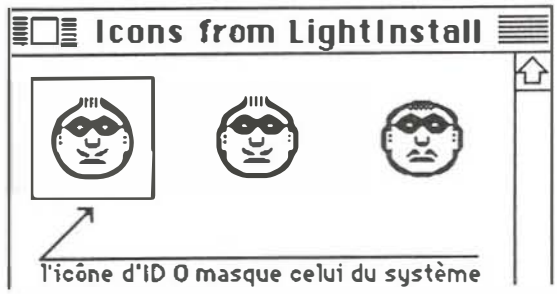

Le dialogue offre cependant l'opportunité de lancer une autre application. Si celle-ci est exploitée, le dialogue standard de choix d'un fichier apparaît, et permet de choisir l'application à lancer. Si celle-ci se trouve dans le même dossier que "LightInstall", son nom s'écrit dans la ressource de type 'STR ' et d'ID 172 et elle devient l'application lancée automatiquement par "LightInstall".

Pour changer le nom de l'application à lancer par "LightInstall", deux possibilités :

- à l'aide d'un éditeur de ressources, changer le nom dans la ressource de type 'STR ' et d'ID 172 ;
- presser le bouton de la souris lors du lancement de l'application "LightInstall" ; le dialogue déjà cité ci-dessus apparaît alors, et l'on peut procéder comme plus haut.

Dans tous les cas, "LightInstall" et l'application qu'elle doit lancer automatiquement doivent se trouver dans le même dossier.

Il est également possible de modifier les icônes de l'application à l'aide de ResEdit :




*N.D.L.R. : l'accessoire "Corner Ligth" livré sur la disquette Macintosh 35 est configuré pour détecter les accès disques et AppleTalk. Si vous désirez modifier la configuration et ne disposez pas de l'éditeur de ressources "ResEdit", il est disponible sur la disquette 'Mac G' (bon de commande page 71).*

## Source 'Corner light.Asm'

```
*                    ---------------
*                    - CORNER LIGHT -
*                    ---------------

*   indicateur d'accès disque
*   ou autre driver

;-----------------------------------------------------------------------------
*   La routine d'ouverture installe
*   une routine à exécuter pendant la remontée du faisceau
*   et collecte la fréquence d'exécution dans les ressource
*
*   La routine de contrôle est vide
*
*   La routine de fermeture
*   - détruit le handle installé dans le DCE,
*   - enlève la routine du VRM
*   - enlève le repère au-dessus de la pomme
*   - détruit le handle des index d  drivers installés dans
*       le heap du système
*-----------------------------------------------------------------------------
*------------------------------- PRÉLIMINAIRES -------------------------------
*------------------------------------ INCLUDES -------------------------------
includeQuickEqu.D
include ToolBEqu.D
include SystEqu.D
includeMacTraps.D
includeMABcro.TXT
*------------------------------------ EQUATES --------------------------------
vblTPtrSize     equ 14      ; taille de vblTaskPtr
indic           equ $0100   : marque au-dessus de la pomme
*---------------------------------- affiche ----------------------------------
numIcon         equ 1
numRect         equ 2

*------------------------------------ X --------------------------------------
        xDef    DERU
*-----------------------------------------------------------------------------
        DERU :
*------------------------------------ MOTS D'ENTRÉE --------------------------
*---------------------------------- comportement -----------------------------
```

```
    dc      $2400   ;réponse aux contrôles et actions périodiques
    dc      30000   ;long délai car contrôle inutile
    dc      0       ;pas d'événements
    dc      0       ;pas de menu
*------------------------- Écarts vers les Routines ------------------------
    dc      DIGORADYR-DERU ; 1 : ouverture
    dc      Eshy-DERU      ; prime (pas utilisé)
    dc      YRZ-DERU       ; 2 : controle
    dc      Eshy-DERU      ; status (pas utilisé)
    dc      KLOZADYR-DERU  ; 3 : fermeture
*------------------------------- Titre -------------------------------------
talbenndc.B   3
    dc.B    'DDO'          ; pour savoir s'il est là
    .ALIGN 2
*------------------------------ OUVERTURE ----------------------------------
*   A3 : DCEPtr
*   D3 : n°, puis ID de l'accessoire
*----------------------------------------------------------------------------
DIGORADYR
*---------------------------- PRELIMINAIRES -------------------------------
    moveA.L     A1,A3       ; DCEPtr
*------------------------------ ACCESSOIRE OUVERT ? -------------------------
;   on examine si le DCE contient un handle
;****************************************************************************
    tst.L       dCtlStorage(A3)
    beq.S       -10     ; = construction
    emp         .W,dCtlRefNum(A3)
    _CloseDeskAcc
    bra Eshy
*------------------------------ OUVERTURE ----------------------------------
-10
*------------------------------ ID de Base ----------------------------------
    move        dCtlRefNum(A3),D3 ;n° de l'accessoire
    addQ        #1,D3
    neg         D3          ;ID de l'accessoire
    lsl         #5,D3
    or          #$C000,D3   ;ID de base = D3
*------------------------------------- BRANCHEMENT ------------------------
    move        Dija,D0     ;première ouverture ?
    beq         -50         ;oui
*------------------------------ AFFICHE ------------------------------------
-20
*------------------------------- Dialogue ----------------------------------
    ep          L
    emp         .L,SP
    _GetPort
*------------------------------ ID de l'icone -------------------------------
    GetRes      #'DITL',D3
    dep         A.L,A0
    moveA.L     (A0),A0
    move        D3,Wo+itmData+By(A0)
*------------------------------ collecte ------------------------------------
    ep          L           ;pour le DialogPtr
    emp         .W,D3
    emp         .L,#0       ; = heap
    emp         .L,#-1      ;fenêtre de premier plan
    _GetNewDialog           ;fenêtre fabriquée !
    moveA.L     (SP),A2     ;DialogPtr = A2 & pile
*------------------------- révélation et décoration -------------------------
; DialogPtr déjà sur la pile
    _ShowWindow
    emp         .L,A2
    _SetPort
    pea         WindowPort+PortRect(A2)
    PaintRect
*------------------------------- nom de l'accessoire ------------------------
*------------------------------- placement du curseur -----------------------
*   On prend un coin du rectangle du dialogue
*   Rectangle = pile
;****************************************************************************
    subA        #Wo+Lo+LR,SP    ;itemType/itemHandle/itemBox
    emp         .L,A2           ;DialogPtr
    emp         .W,#numRect     ;n° de l'item
    pea         Wo+Lo(SP)       ;type
    pea         Lo+Wo+Lo+Wo(SP)         ;handle
    pea         2*Lo+Wo+Lo+Wo+Lo(SP)    ;rectangle
    _GetDItem
    addA        #Wo+Lo+Lo,SP    ;on laisse un coin du
                                rectangle sur la pile
    _MoveTo                     ;RAN
*------------------------------ collecte ------------------------------------
    subA        #Wo+Lo+LStr,SP      ;nom/type/ID
    emp         .L,dCtlDriver(A3)
    pea         Lo+LStr+Lo(SP)      ;ID
    pea         2*Lo+LStr(SP)       ;type
    pea         3*Lo(SP)            ;nom
    _GetResInfo
*------------------------------ écriture ------------------------------------
    emp         .W,#boldV+outlineV      ;+shadowV
    ;style
    _TextFace
    emp     .L,SP                   ;adresse du nom
    _DrawString                     ;tracé
    addA        #Wo+Lo+LStr,SP      ;RAN
*------------------------------ autres libellés -----------------------------
    emp         .W,#0
    _TextFace                       ;suppression des styles
    SetFont     #geneva,#9          ;fonte pour les libellés
    emp         .L,A2
    _DrawDialog
*------------------------- fermeture de l'affiche ---------------------------
    GetRes      #'pds#',D3
    dep         A.L,A1
    moveA       .L  (A1),A1
    emp         .W,D3   ;sauvegarde
    clr.L       D3
    move        Wo(A1),D3
    add.L       Ticks,D3

-30 cmp.L       Ticks,D3
    bmi.S       -40         ;délai atteint
    ep          W
    _Button
    dep         .W,D0
    beq.S       -30         ;souris en haut

-40 dep         .W,D3       ;restauration
    emp         .L,A2
    _DisposDialog
    _SetPort
*------------------------- INSTALLATION DU DELAI ----------------------------
*   branchement ici pour la première ouverture
;****************************************************************************
-50 mouv0       W,#1.Dija       ;mise à jour de l'indicateur
    GetRes      #'pds#',D3
    moveA.L     (SP),A0         ;handle
    moveA.L     (A0),A0         ;adresse
    mouv1       W,(A0),pad0
    _ReleaseResource
*------------------- INSTALLATION DES INDEX DE DRIVERS ----------------------
*   placement dans un handle dans l'entrepot du système
;****************************************************************************
    GetRes      #'NDr#',D3
    moveA.L     (SP),A2         ;on garde le handle sur la pile
    ep          L               ;pour la taille
    emp         .L,A2           ;handle sur la pile
    _SizeRsrc
    dep         .L,D0           ;taille en octets
    moveA.L     (A2),A2         ;adresse
    move.L      D0,D1           ;sauvegarde
    _NewHandle ,SYS             ;création
    mouv1       L,A0,NoDriv     ;enregistrement dans la zone de données
    moveA.L     (A0),A1         ;pour la destination
    moveA.L     A2,A0           ;origine
    move.L      D1,D0           ;taille du transfert
    _BlockMove
    lsr         #1,D1           ;nombre de drivers à tester
    mouv0       W,D1,NbrDriv    ;enregistrement dans la zone de données
    ;ResHandle sur la pile
    _ReleaseResource
*------------------- INSTALLATION DE LA ROUTINETTE ----------------------
*------------------------------ Création du Pointeur ------------------------
    lea         Vertige,A0              ;origine
    lea         Chute,A1                ;extrémité
    subA.L      A0,A1                   ;différence
    move.L      A1,D0                   ;taille du bloc
    _NewHandle ,SYS
    _HLock
    move.L      A1,D0                   ;restauration
    move.L      A0,dCtlStorage(A3)      ;sauvegarde du handle
*------------------------------ Transfert -----------------------------------
    moveA.L     (A0),A0             ;le pointeur construit...
    moveA.L     A0,A1               ;...devient destination
    lea         Vertige,A0          ;origine
    _BlockMove                      ;transfert
*------------------------------ Installation --------------------------------
    ;adresse de la routine
    lea         ROUTINETTE-Vertige(A1),A0
    move.L      A0,vblAddr(A1)          ;placement
    moveA.L     A1,A0                   ;adresse de vblTask
    _VInstall                           ;installation
    bra.S       Eshy
*------------------------------ FERMETURE -----------------------------------
KLOZADYR
*------------------- ENLEVEMENT DE LA ROUTINETTE ET DIVERS ------------------
    moveA.L     dCtlStorage(A1),A0      ;handle
    moveA.L     (A0),A0                 ;adresse
    _VRemove                            ;enlèvement de la routine
;destruction des handles
    moveA.L     dCtlStorage(A1),A0
    _DisposHandle
    moveA.L     NoDriv,A0
    _DisposHandle
;effacement de l'indicat on au dessus de la pomme
    moveA.L     ScrnBase,A0
    move        #0,Wo(A0)
;pensons à la réouverture
    clr.L       dCtlStorage(A1)
    bra.S       Eshy
;------------------------------ CONTROLE ------------------------------------
*                   le controle est vide
;----------------------------------------------------------------------------
YRZ
*----------------------------------------------------------------------------
    emp     .L,jIODone ;"control request" traité, saut à ioDone
*---
Eshy
*----------------------------------------------------------------------------
    moveQ       #0,D0   ; return no error
```

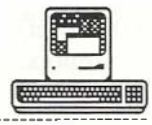
```
                    ;Kenavo
        RTS
*------------------------------ SOUS-ROUTINES ------------------------------
*---------------------------------------------------------------------------
*------------------------------ POUR LA ROUTINETTE -------------------------
Vertige
    dc.L        0   ;suivant
    dc          1   ;type de la queue
    dc.L        0   ;adresse de la routine
Padez
    dc          9   ;délai entre les actions (tiernes)
    dc          0   ;sans usage
;données
Pad0
    dc          0   ;délai initial
NbrDriv
    dc          0   ;nombre de drivers à tester
NoDriv
    dc.L        0   ;handle des index de drivers
*---------
ROUTINETTE
*------------------------------ ROUTINE A INSTALLER ------------------------
; Cette routine, destinée à être exécutée pendant la remontée du faisceau,
; examine si le driver de disque est en service et, si oui, l'indique en
; exécutant la routine placée plus bas
*------------------------------ PRELIMINAIRES ------------------------------
    lea         *,A0    ; adresse 'ROUTINETTE'
;réinitialisation du délai
    move        Pad0-ROUTINETTE(A0),Padez-ROUTINETTE(A0)

    moveA.L     ScrnBase,A2     ;au cas où la marque aurait été…
    orI         #indic,No(A2)   ;…effacée au-dessus de la pomme

    moveA.L     NoDriv-ROUTINETTE(A0),A3 ;handle des index de drivers
    moveA.L     (A3),A3                  ;adresse
    move        NbrDriv-ROUTINETTE(A0),D1 ;nombre de drivers (compteur)
*------------------------------ TEST DES DRIVERS ---------------------------
    moveA.L     UTableBase,A0       ;adresse de la UnitTable

-0  move        (A3)+,D0            ;index du driver
    tst.L       0(A0,D0)            ;handle du driver
    beq.S       -5                  ;non en service (handle nul)
    moveA.L     0(A0,D0),A1
    moveA.L     (A1),A1
    btst        #drvrActive,dCtlFlags+By(A1)
    bne.S       -20                 ;driver en service
-5  subQ        #1,D1               ;décrémentation du compteur
    bne.S       -0
*------------------------------ CA NE TOURNE PAS ---------------------------
    btst        #7,(A2)         ; Si c'est déjà noir,…
    bne.S       -10             ;…sortie, sinon…
    orI         #$C000,(A2)     ;…on noircit…
    addA        ScrnRow,A2      ;…où il…
    orI         #$8000,(A2)     ;…faut.
-10 RTS
*------------------------------ CA TOURNE ----------------------------------
-20 eorI        #$C000,(A2)
    addA        ScrnRow,A2
    eorI        #$8000,(A2)
    RTS

Chute
*------------------------------ DONNÉES ------------------------------------
Déja
    dc  0       ;a déjà été ouvert ou non
*---------------------------------------------------------------------------
                FIN
---------------------------------------------------------------------------
```

## Source 'Corner Install'

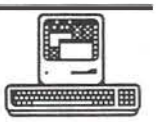
```
*                 ---------------
*                 - APPDISKATOR -
*                 ---------------
*   application lançant l'accessoire de bureau "Corner Light"

*---------------------------------------------------------------------------
;"Embellissement" de l'écran à l'aide d'un motif aléatoire.
;Le GrafPtr n'est pas détruit, ainsi que certaines autres ressources,
;car l'application débouche sur le lancement d'une application sans pouvoir
;revenir en arrière.
;accessoire :
;Recherche de l'accessoire par son nom interne.
;En cas d'absence, affichage d'un dialogue éphémère.
;Sinon, ouverture de l'accessoire.
;application à lancer :
;Recherche de l'application dont le nom est donné dans la ressource de type STR .
;En cas d'absence, affichage d'un dialogue permettant d'en choisir une autre.
;Lorsque cette nouvelle application est lancée, son nom est enregistré dans
;la ressource.
;Si l'application initialement prévue est trouvée, elle est lancée.
*La liste des délais n'est pas ouverte au début,
*car en fonctionnement normal son ouverture est inutile.
*---------------------------------------------------------------------------
*   A1 : handle de l'accessoire
*   A2 : GrafPtr
*        StringHandle du nom de l'accessoire
*        StringHandle du nom de l'application
*   A3 : DeskPattern
*        SFReply
*        EventRecord
*   A4 : DialogPtr
*   D3 : compteur pour le motif
*        délai du dialogue en cas d'absence de l'accessoire
*------------------------------ HORS D'OEUVRE ------------------------------
*------------------------------ INCLUDES -----------------------------------
include     MacTrape.D
include     QuickEqu.D
include     ToolBEqu.D
include     FilEqu.D    ;equates des fichiers
include     SystEqu.D   ;equates du système
include     mABcro.Txt
*------------------------------ EQUATES ------------------------------------
abID            equ     $AB         ; 171
nomAcc          equ     $0344444F   ; nom interne de l'accessoire (DDO)
padez           equ     60
DeskPattern     equ     $A3C        ; desk pattern [8 bytes]
accAbsIc        equ     3
appAbsIc        equ     4
*------------------------------ Divers -------------------------------------
infErr          equ     -43     ;fichier introuvable
drvrName        equ     18      ;nom de l'accessoire [Str255]
hollDegwez      equ     $0000FFFF
*------------------------------ X ------------------------------------------
*------------------------------ xRef ---------------------------------------
    xRef    CollFich
    xRef    SetDefDisk
*------------------------------ xDef ---------------------------------------
    xDef    DEBU
*---------------------------------------------------------------------------
    DEBU :
*------------------------------ INITIALISATIONS ----------------------------
    pea     GrafGlobals(A5)     ;variables globales de QuickDraw
    _InitGraf
    _InitFonts
    _InitWindows
    _InitMenus
    _teInit
    cmp     .L,#0               ; No restart procedure
    _InitDialogs
    _InitCursor
*------------------------------ OUVERTURE DE L'ACCESSOIRE ------------------
*------------------------------ DECORATION ---------------------------------
    move.L      #portRec,D0
    _NewPtr
    moveA.L     A0,A2
    cmp         .L,A2
    _OpenPort
*------------------------------ peinture -----------------------------------
    moveA.L     (A5),A0     ;pointeur sur les variables globales de QuickDraw
    move.L      Time,Randseed(A0)       ;initialisation
    move.L      #DeskPattern,A3
    moveQ       #4,D3                   ;compteur
-0  cp          W
    _Random
    dep         .W,(A3)+
    subQ        #1,D3
    bne.S       -0

    pea         WindowPort+PortRect(A2)
    cmp         .L,#DeskPattern         ;adresse du motif
    _FillRect
*------------------------------ NOM DE L'ACCESSOIRE ------------------------
    CallRes     #abID,String
    dep         A.L,A2      ;handle du nom obligé de l'accessoire
*------------------------------ RECHERCHE ----------------------------------
    cp          W
    cmp         .L,#'DRVR'
    _CountResources
    dep         .W,D1       ;nombre de drivers dans les fichiers ouverts
-10 cp          L
    cmp         .L,#'DRVR'
    cmp         .W,D1
    _GetIndResource
    moveA.L     (SP),A1             ;ResHandle
    moveA.L     (A1),A0             ;adresse du driver
    move.L      drvrName(A0),D0     ;on cherche un nom de 3 lettres (DDO)
    cmpI.L      #nomAcc,D0
    beq         OuvrAcc
    ;ResHandle déjà sur la pile
    _ReleaseResource
    subQ        #1,D1
    bne.S       -10
*------------------------------ ACCESSOIRE ABSENT --------------------------
*------------------------------ Dialogue -----------------------------------
    cmp         .L,(A2) ;nom de l'accessoire
    cmp         .L,#0
    cmp         .L,#0
    cmp         .L,#0
    _ParamText
    bsr.W       COLLNOMAPP
    move        #abID,D0
    bsr.W       AFFDIAL
*------------------------------ Attente ------------------------------------
    moveQ       #0,D1
    bsr.W       COLLOEL
-20 cmp.L       Ticks,D3
    bmi.S       -30     ;délai atteint
    bsr.W       TEST
    beq.S       -20     ;souris en haut
-25 bsr.W       TEST
```

```
    bne.S     -25     ;souris en bas
*------------------------------ Epilogue ------------------------------
-30 moveQ     #accAbsIc,D0   ;n° de l'item
    bsr.W     FURAX
    bra.S     LANCAPP
*------
OuvrAcc
*------------------------------ ACCESSOIRE PRESENT ------------------------------
*------------------------------ Manipulation du Nom ------------------------------
*          le handle du nom forcé est dans A2
*------------------------------ collecte de l'ID ------------------------------
    subA      #LStr+Lo+Wo-Lo,SP ;nom/type/ID, - ResHandle
    emp       .L,A1              ;handle de l'accessoire
    pea       Lo+LStr+Lo(SP)     ;ID
    pea       2*Lo+L tr(SP)      ;type
    pea       3*Lo(SP)           ;nom
    _GetResInfo
    move      LStr+Lo(SP),D1     ;ID
    addA      #LStr+Lo+Wo,SP     ;RAN
*------------------------------ changement du nom ------------------------------
    emp       .L,A1
    emp       .W,D1
    emp       .L,(A2)
    _SetResInfo
    emp       .L,A1
    _ChangedResource
    emp       .L,A1
    _WriteResource
*------------------------------ Est-il Ouvert ? ------------------------------
    lsl       #2,D1              ;multiplication par 4
    moveA.L   UTableBase,A0      ;adresse de la UnitTable
    moveA.L   0(A0,D1),A0        ;handle du DCE
    cmp.L     #0,A0
    beq.S     -10                ;l'accessoire n'a pas encore été ouvert
    moveA.L   (A0),A0            ;adresse du DCE
    tst.L     dCtlStorage(A0)    ;routinette installée ?
    beq.S     -10                ;non
    bra.S     LANCAPP            ;oui
*------------------------------ Ouverture ------------------------------
-10 ep        W                  ;pour le n°
    emp       .L,(A2)            ;nom
    _OpenDeskAcc
    clr       (SP)+              ;enlèvement du n°
*------
LANCAPP
*------------------------------ LANCEMENT D'UNE APPLICATION ------------------------------
    bsr.W     COLLNOMAPP         ;si ce n'est déjà fait
*------------------------------ VERIFICATION DE LA PRESENCE ------------------------------
    move.L    #ioFUelSize,D0
    _NewPtr   ,CLEAR
    _GetVol   ;volume par défaut
    move.L    (A2),ioFileName(A0)   ;adr. du nom de l'appl. = bloc
    _GetFileInfo
    SSi       I.W.#fnfErr,D0,AppAbs
*------------------------------ LANCEMENT ------------------------------
*------------------------------ Dernière Chance ------------------------------
    bsr.W     TEST
    beq.S     -10
-0  bsr.W     TEST
    bne.S     -0
    bra.S     AppAbs
*------------------------------ On y va ------------------------------
-10 ep        W          ;main sound and screen buffers
    emp       .L,(A2)    ;adresse du nom
    moveA.L   SP,A0
    _Launch
*------
AppAbs
*------------------------------ ABSENCE : CHOIX D'UNE AUTRE APPLICATION ------------------------------
*Un dialogue permet de lancer une autre application
*ou de retourner au finder.
*En cas de silence de l'utilisateur,
*on retourne au finder passé un certain délai, d'où une boucle d'évènement.
*------------------------------ Affichage du Dialogue ------------------------------
    move      #abID+1,D0
    bsr.W     AFFDIAL
*------------------------------ Discussion ------------------------------
*------------------------------ initialisations ------------------------------
    clr.L     D0
    move      #evtBlkSize,D0
    _NewPtr
    moveA.L   A0,A3   ;EventRecord, garder le handle du nom dans A2
    moveQ     #1,D1   ;second délai
    bsr.W     COLLDEL
    move.L    #hollDegrez,D0
    _FlushEvents
*------------------------------ boucle ------------------------------
-0  cmp.L     Ticks,D3
    bpl.S     -1
*------------------------------ délai écoulé ------------------------------
    moveQ     #appAbsIc,D0   ;n° de l'item
    bsr.W     FURAX
    bra.S     -10            ;retour au finder
*------------------------------ délai non écoulé ------------------------------
-1  ep        W
    emp       .W,#hollDegrez     ;masque
    emp       .L,A3              ;adresse de l'enregistrement de l'évènement
    _GetNextEvent
    dep       .W,D0
    beq.S     -0
    move      evtNum(A3),D0
    cmp       #mButDwnEvt,D0
    bne.S     -0
    subA      #Lo+Wo,SP          ;DialogPtr/ItemHit
    ep        W
    emp       .L,A3
    pea       Lo+Wo(SP)          ;DialogPtr
    pea       2*Lo+Wo+Lo(SP)     ;ItemHit
    _DialogSelect
    dep       .W,D0
    beq.S     -0
*------------------------------ sortie ------------------------------
    moveQ     #appAbsIc,D0       ;n° de l'item
    bsr.W     FURAX
    clr.L     (SP)+              ;enlèvement du DialogPtr
    dep       .W,D0              ;ItemHit, RAN
    SSi       I.W,#okButton,D0,AutrApp
*------------------------------ retour au finder ------------------------------
-10 emp       .L,A2
    pea       FinderName
    _SetString
    emp       .L,A2
    _ChangedResource
    _ExitToShell
*------
AutrApp
*------------------------------ Lancement d'une autre Application ------------------------------
*  le handle du nom est toujours dans A2
*------------------------------ DLOG -4000 ------------------------------
    GetRes    #'coin',#abID
    dep       A.L,A0
    moveA.L   (A0),A0
    move.L    (A0),D1
    emp       .L,#'APPL'
    moveA.L   SP,A0          ;adresse de la liste des types
    subA      #74,SP         ;pour le sfReply
    moveA.L   SP,A1          ;adresse de la liste du sfReply
    moveQ     #1,D0          ;un seul type
    bsr.W     CollFich
    tst D0
    bne.S     -10            ;pas d'annulation
    addA      #74+Lo,SP      ;enlèvement du sfReply et du type
    _ExitToShell             ;en cas d'annulation
*------------------------------ vérification ------------------------------
;          si on relance l'application, on boucle
;*****************************************************************************
-10 moveA.L   A1,A3            ;sauvegarde
    clr.L     D0
    moveA.L   #CurApName,A0    ;adresse
    move.B    (A0),D0          ;longueur
    addA      #1,A0
    swap      D0
    lea       rName(A3),A1     ;adresse
    move.B    (A1),D0          ;longueur
    addA      #1,A1
    _CmpString ,CASE
    tst.L     D0
    bne.S     -20              ;différent
*------------------------------ choix de l'application en cours ------------------------------
    pea       CurApName
    emp       .L,#0
    emp       .L,#0
    emp       .L,#0
    _ParamText
    ep        W
    emp       .W,#abID+2
    emp       .L,#0
    _StopAlert
    dep       .W,D0
    bra.S     AutrApp
*------------------------------ enregistrement du nom ------------------------------
-20 emp       .L,A2
    pea       rName(A3)
    _SetString
    emp       .L,A2
    _ChangedResource
*------------------------------ lancement ------------------------------
    move      rVolume(A3),@1     ;RefNum du disque
    bsr.W     SetDefDisk
    ep        W                  ;main sound and screen buffers
    pea       rName(A3)
    moveA.L   SP,A0
    _Launch
*------------------------------ SOUS-ROUTINES ------------------------------
COLLNOMAPP
*------------------------------ COLLECTE DU NOM DE L'APPLICATION ------------------------------
    CollRes   #abID+1,String
    dep       A.L,A2     ;handle, adresse utilisée plus bas
    emp       .L,#0
    emp       .L,(A2)    ;adresse du nom de l'application
    emp       .L,#0
    emp       .L,#0
    _ParamText
    RTS
*------
AFFDIAL
*------------------------------ AFFICHAGE DE DIALOGUE ------------------------------
* entrée : ID - D0
* sortie : A4 - DialogPtr
;*****************************************************************************
    emp       .W,D0      ;sauvegarde
    emp       .W,#1
```

corner light



```
    _SysBeep
    dep         .W,D0       ;restauration
    ep          L
    emp         .W,D0
    emp         .L,#0
    emp         .L,#-1
    _GetNewDialog
    moveA.L     (SP),A4
    _DrawDialog
    RTS
*------    ---
COLLDEL
;------------------------- COLLECTE D'UN DELAI -------------------------
; entrée : n° du délai (entre 0 et 2) = D1
; sortie : D3 = date de fin
;*****************************************************************
    lsl         #1,D1           ;multiplication par 2
    GetRes      #'pdz#',#abID
    dep         A.L,A0          ;handle de la liste des délais
    moveA.L     (A0),A0         ;adresse
    clr.L       D3
    move        (A0,D1),D3      ;délai en tierces
    add.L       Ticks,D3
    RTS
*----       ---
FURAX
*-------------------- EMBELLISSEMENT AVANT DISPARITION --------------------
*entrée : DialogPtr = A4
*     n° de l'item = D0
;*****************************************************************
    emp         .W,D0               ;sauvegarde
    emp         .L,A4
    _SetPort
    dep         .W,D0               ;restauration
    subA        #LR+Lo+Wo,SP        ;Rect/ItemHandle/ItemType
    emp         .L,A4               ;DialogPtr
    emp         .W,D0               ;n° de l'item
    pea         Wo+Lo+LR+Lo(SP)     ;type
    pea         Lo+Wo+Lo+LR(SP)     ;handle
    pea         2*Lo+Wo+Lo(SP)      ;rectangle
    _GetDItem
    emp         .L,SP               ;adresse du rectangle
    ep          L                   ;place pour le handle
    emp         .W,#abID+1          ;ID
    _GetIcon
    ;tout est en place
    _PlotIcon
    addA        #LR+Lo+Wo,SP        ;RAN
    GetRes      #'pdz#',#abID
    dep         A.L,A0              ;handle de la liste des délais
    moveA.L     (A0),A1
    moveA.L     #0,A0
    move        2*Wo(A1),A0
    _Delay
    emp         .L,A4               ;DialogPtr
    _DisposDialog
    RTS
*---       ---
TEST
*------------------------ ETAT DU BOUTON DE LA SOURIS ---------------------
    ep          N
    _Button
    dep         .W,D0
    RTS
*------------------------------- CONNÉES ------------------------------
*   Néant
*-----------------------------------------------------------------------
    FIN
```



# En vitrine

## Numérisation

**Datacopy** (20 920 F HT) est un outils de digitalisation (tripotage ?) qui se raccorde au port SCSI du Macintosh. Il traite une page au format A4 en 15 secondes et est aussi disponible pour IBM (personne n'est parfait...). Distribué par **Alpha Systèmes**.

## DAO de gamme

**VersaCAD** (18 900 F HT) est un logiciel de dessin assisté par ordinateur qui sait faire plein de choses ; comme il n'est pas question de décrire en détails un tel programme dans cette rubrique, le mieux est encore de demander une documentation (ou une démonstration) à votre revendeur. Revendeur qui, s'il le désire, peut participer à un séminaire de formation à VersaCAD pour 4 000 F HT. Distribué par **Alpha Systèmes**.

## Kodacintosh

**SlideWriter+ d'Honeywell** (entre 85 000 et 90 000 F HT quand même) réalise des diapositives 24x36 à partir des images couleurs du Mac II, sur Kodak Ektachrome 100 ISO ou Polaroïd Polachrome 100 ISO. La bête (dont le cœur est un MC68000) a une résolution de 4096 pixels par ligne et peut pondre 45 dias par heure. Si vous êtes moins pressés, vous pourrez vous en sortir avec votre boîtier reflex habituel. Une version Macintosh Plus/SE est actuellement à l'étude.

## Câbles twinaxiaux

**Belden Electronics** propose des nouveaux câbles 78Ω pour le réseau AppleTalk : le Belden 9999 est isolé avec du polyéthylène alors que le Belden l'est avec du Teflon FEP, ce qui autorise une installation en faux-plafond sans gainages supplémentaires. De bien beaux câbles...

## Écriture totale

**FullWrite Professional** (£339) de **Computer Unlimited**, est un de ces nouveaux logiciels de traitement de textes qui offrent des possibilités auxquelles on n'aurait même pas osé penser il y a seulement trois ou quatre ans. Celui-ci intègre un système de traitement des idées du type 'Think Tank", les césures automatiques bien sûr, une option 'Get info' qui donne la taille du document traité ainsi que le nombre de mots qu'il contient, une possibilité de modification du 'kerning' (espace entre les lettres), un dictionnaire de 100 000 mots (anglais pour l'instant), une recherche avec style, taille, etc., une possibilité de création 'd'objets graphiques' du type "MacDraw" sans passer par un tel programme, etc.

## Écran total

Les **StarCrex** 19 et 20 fabriqués et distribués par **Crex technology**, sont des écrans monochromes haute résolution pour Mac SE. Ils permettent l'affichage de deux pages au format A4. La carte interface comporte 128Ko de RAM et se loge dans le 'slot' du SE.

## Mac ↔ IBM

**MacLink Plus** (2 550 F TTC) est conçu pour les transferts de fichiers

entre Macintosh et IBM ou entre Macintosh et Macintosh en local ou par le réseau téléphonique. Points faibles par rapport à InterPom's 2.0 : il ne permet pas l'utilisation du Minitel (modem obligatoire), ne semble pas se charger du recodage des caractères accentués et spéciaux, ne permet pas la communication avec un Apple // et est beaucoup plus cher. Mise à jour MacLink/MacLink Plus : 500 F TTC Distribué par **Alpha Systèmes**.

## Apple éducation

**Apple Computer France** vient de signer une Convention avec l'UNAPEI (Union Nationale des Associations de Parents d'Enfants Inadaptés) pour favoriser le développement de l'utilisation de la micro-informatique au service des personnes handicapées mentales.

## AppleTalk téléphonique

Il est désormais possible de connecter un Macintosh 'hors site' sur AppleTalk via le réseau téléphonique. Deux possibilités : **NetSerial** (3 950 F HT) si vous disposez déjà d'un modem aux normes Hayes ; **NetModem** (6 950 F HT) si vous n'avez pas ledit modem. Les 'softs' sont compatibles Tops, AppleShare ou encore MacServe mais ne permettent pas l'utilisation du Minitel comme modem. Distribué par **Alpha Systèmes**.

## Projetez le Mac...

...mais seulement l'image de son écran avec le rétroprojecteur **MacWiewFrame**. L'appareil est livré dans une valise – donc facilement transportable –, a une résolution de 640 x 400 points (trop juste pour le Mac II), permet la projection en inverse (blanc sur noir) et coûte 16 900 F HT. Distribué par **Alpha Systèmes**.

## Système expert

**NEXPERT Objet** (50 000 F HT) est un système expert très évolué conçu par **Neuron Data Inc** et distribué en France par **Intellia**. Le prix comprend l'environnement de développement complet avec documentations, plus une demie journée de prise en main, une assistance téléphonique et les mises à jour du logiciel pendant une durée de 6 mois. Intellia est aussi le concepteur de **N-Toolkit** (21 000 F HT), une boîte à outils pour NEXPERT.

## Sécurité

**Irwin Magnetics**, surtout connu pour ses mémoires de masse sur PC, vient au monde Macintosh en annonçant l'arrivée prochaine de systèmes de sauvegarde pour 10, 20, 40 ou 64 Méga-octets sur mini-cartouches.

## Sécurité encore

**Infosec**, le salon de la sécurité informatique organisé par **MCI**, se tiendra les 7, 8 et 9 juin 1988 au Palais des congrès de Paris

## Grammaire anglaise

**Sensible Grammar** ($99.95) de **Sensible Software**, est la version Macintosh d'un logiciel de vérification de textes qui existe déjà depuis plusieurs années pour les Apple //. Il ne fonctionne bien sûr pas avec les textes en français.

## Pour Macintosh baladeur

**Kangourou services** fabrique et distribue **MAC.CASE**. Il s'agit d'un bagage conçu pour transporter un Macintosh Plus ou SE, un disque dur et le lecteur externe. On peut aussi obtenir un bagage assortie permettant le transport d'une imprimante ImageWriter II. Ces articles peuvent également être obtenus auprès des revendeurs Apple. Les prix : bagage pour Mac décor loupe d'orme : 1 215 F HT ; bagage pour Mac décor pointe de diamant : 1 125 F HT ; bagage pour ImageWriter II décor loupe d'orme : 1 079 F HT ; bagage pour ImageWriter II décor pointe de diamant : 975 F HT.

## MacWrite 5.0

Cette nouvelle version nous arrive avec un vérificateur orthographique intégré et un dictionnaire de 100 000 mots. De plus, **MacWrite 5.0** offre des 'raccourcis clavier', une commande 'tout sélectionner', la commande du curseur avec ←↑→↓, la possibilité d'utiliser pleinement le MultiFinder et les écrans de grande taille. Le logiciel fonctionne sur les Macintosh 512/800Ko, Plus, SE et II. 1 190 F HT ; mise à niveau : 500 F HT. Distribué par **Claris/P-Ingénierie**.

## MacPaint 2.0

Nouvelle version totalement refondue du plus célèbre programme pour Macintosh. On ne peut plus maintenant lui reprocher grand chose : jusqu'à neuf fenêtres simultanément, menus 'détachables', défilement automatique, affichage 'pleine page' (20,3 x 25,4 cm sur un grand écran), possibilité de sauvegardes du document à différents stades du travail, affichage des tailles et position de la forme traitée, travail sur des agrandissements à 200, 400 ou 800%, ou encore sur une réduction à 50%... Fonctionne sur les Macintosh 512/800Ko, Plus, SE et II. 990 F HT ; mise à niveau : 500 F HT. Distribué par **Claris/P-Ingénierie**.

## MacDraw II

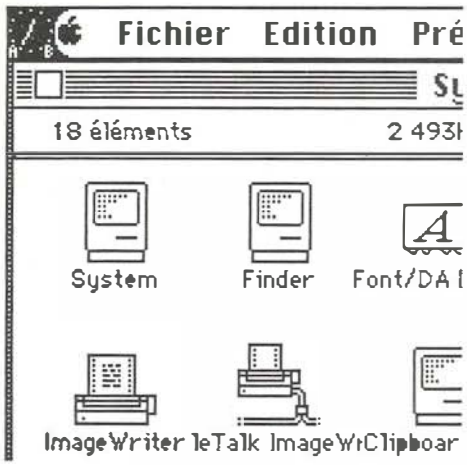
Beaucoup plus rapide que ses versions antérieures (de 3 à 10 fois selon les opérations), MacDraw gère désormais la couleur et est surtout devenu plus précis : 2000 points par pouce, hauteur et largeur du crayon réglables par incréments de 0,001 pouces centimètres ou points, réductions ou agrandissements de 3,12 à 3200%... MacDraw II fonctionne sur les Macintosh 512/800Ko, Plus, SE et II. 3 490 F HT ; mise à niveau : 1 250 F HT. Distribué par **Claris/P-Ingénierie**.

*Vous trouverez les adresses des distributeurs page 70* ➞

# Toujours disponible :

Alain Bohec

# Mini-montre analogique




L'affichage de l'heure sur le Macintosh a déjà été traité souvent et de multiples façons mais – et c'est là chose étrange – presque toujours avec la solution numérique. Or, et bien qu'il y ait certainement autant de partisans de l'affichage numérique que de l'affichage analogique, il semble que la seconde méthode permette une lecture plus instinctive, même si elle n'autorise pas la même précision.

L'emplacement retenu – à gauche de la ⌘ – étant extrêmement exigu (20/20 points), la méthode analogique s'imposait du fait de l'impossibilité d'afficher un minimum de quatre chiffres dans un espace aussi réduit.

Voici donc une horloge toujours lisible, qui n'est pas perturbée par les menus comme le sont certaines montres numériques qui se placent dans la barre de menu, mais à droite, et qui occupera très peu de place sur vos disques et disquettes (873 octets).

## Source 'MiniMontre.Asm'

```
; MiniMontre.Asm

*                ...............
*                · MiniMontre ·
*                ...............
*            MONTRE EN HAUT A GAUCHE
*-----------------------------------------------------------
* Le pointeur sur le DCE est sauvegardé sur la pile
*          utilisation de tempus.Rel
* PRELIMINAIRES ......................................
* INCLUDES ...........................................
include SystEqu.D
include QuickEqu.D
include MacTraps.D
include mABcro.Txt
* EQUATES ............................................
csCode  equu    26
Pivot   equ     $00080008       ;pivot
Lga     equ     6               ;long grande aiguille
Lpa     equ     4               ;long petite aiguille
* X ..................................................
* xRef ---------------------------------------------
        xRef    GrandAig
        xRef    PetitAig
        xRef    GetTime
* xDef ---------------------------------------------
                xDef    dt
                xDef    DERU
*-----------------------------------------------------------
                DERU :
* MOTS D'ENTREE ......................................
* COMPORTEMENT--------------------------------------
        dc.W    $2400   ;réponse aux ctrl et actions
        dc.W    100     ;toutes les 100 tierces
        dc.W    0       ;aucun événement
        dc.W    0       ;pas da menu
* ECARTS VERS LES ROUTINES -------------------------
        dc.W    DIGORADYR-DERU  ;ouverture
        dc.W    Distro-DERU     ; prime (unused)
        dc.W    YRZ-DERU        ;contrôle
        dc.W    Distro-DERU     ; status (unused)
        dc.W    KLOZADYR-DERU   ;fermeture
* TITRE ---------------------------------------------
talbenn dc.B    3
        dc.B    'kdr'
        .ALIGN  2
* OUVERTURE @@@@
```

```
        mouv0 L,#0,gPtr ;pour réouverture
        emp .L,bmH
        _ReleaseResource
        mouv0 W,#-1,dt+dtMinute ;évite attente évent
*@@@@@
Distro
* CLASSIQUE ........................................
        moveQ   #0,D0   ; return no error
                        RTS
* CONTROLE @@@@@@
```

# Utilitaire DHGR Double-Graphics

Pascal Cantot

De nombreux utilitaires double-haute résolution ont parus dans diverses revues, Pom's notamment : manipulation de pages DHGR, chargement, sauvegarde, etc. Mais à ma connaissance AUCUN n'offrait ce que l'on peut le plus désirer : un Basic Double haute résolution ! De plus, à côté de la Double haute résolution, il y a un oublié : la double basse résolution, 80 x 48, qui présente l'intérêt d'avoir des pixels carrés et non plus rectangulaires.

J'avais écrit en 1983 un amper-interpréteur, ICARE, qui fut publié dans Pom's et eut par la suite de nombreux descendants, chacun adapté au programme Basic que j'écrivais. Il y eut notamment un amper-interpréteur DHGR, dédié à un grapheur de gestion double-haute résolution qui ne fut jamais complètement achevé. Puis j'ai délaissé le Basic pour le 100% machine, puis pour le Turbo Pascal. Fortuitement, je me suis remis au 6502 et, cherchant un idée de programme j'ai pensé à réécrire ICARE en DHGR. Mais j'ai pensé qu'un programme original, inédit et surtout moins long intéresserait plus Pom's et ses lecteurs. Alors, je suis reparti quasiment de zéro. Le programme tient en 1150 lignes de source 6502 et 2Ko de code objet. 1150 lignes tout de même, car il m'a fallu réécrire la majeure partie de la ROM graphique de l'Apple//. Mais le résultat en valait la peine !

"Double graphics" offre en mode double-résolution les instructions Applesoft suivantes : HGR, HGR2, HPLOT, HCOLOR, GR, PLOT, COLOR, HLIN, VLIN, SCRN, plus d'autres fonctions très utiles comme PRINT, DRAW, LET et DEF pour gérer des caractères graphiques DGR couleurs, mixer graphiques DHGR et caractères 40 ET 80 colonnes, et redéfinir jusqu'à 256 caractères graphiques. Sans oublier quelques instructions de fenêtrage : HOME et CLEAR (cette dernière efface la fenêtre DHGR avec possibilité de demi-teintes), et le tant recherché SCRN double-haute résolution.

Et ce n'est pas tout : "Double graphics" comprend des points d'entrée pour être appelé aussi par un programme machine, comme les routines Applesoft, dont il garde (pour HPLOT, PLOT, etc.) les mêmes protocoles de passage des paramètres. Ainsi, un HPLOT DHGR se fait de la même façon qu'un HPLOT 'normal', excepté que X varie de 0 à 559.

## Carte mémoire

$300-$307 : réservé à "Double graphics" :
- $300 : adresse de la table de caractères
- $304 : taille des caractères : <128 : 80 col. ≥128 : 40 col.
- $305 : si ≥128, la pseudo couleur "INVERSE" est utilisée par HPLOT (seul le bit 7 compte)
- $306 : si ≥128 (bit 7 à 1), FILLZ inverse la fenêtre.

$800 : programme Basic
$2000 : (main + aux) : pageDHGR
$4000 : "Double graphics" (environ 2Ko)
$4900 : (conseillé) : table de caractères (longueur L variable)
$4900+L+1 : LOMEM

**Note importante :** l'adresse mise en $300 représente l'adresse du caractère de code 00. Si vous utilisez les polices de 96 caractères d'Icare ou de Haifa, qui commencent par le caractère 32 (espace), il faut mettre en $300 l'adresse de chargement de la table moins 256 (8*32) si vous ne désirez pas utiliser les caractères 0 à 31. Généralement, les polices utilisées ont 96 ou 128 caractères. Mais DHGR peut gérer jusqu'à 256 caractères. En Basic, ne pas oublier que l'instruction CHR$ ne permet l'envoi que de 128 codes. L'idéal me semble être 128 caractères : 96 'normaux' (codes 32 à 127), plus 32 'graphiques' (codes 0 à 31).

La table des fonctions se trouve en début de programme, dont l'origine est en $4000. Chacune d'entre elles est désignée successivement par son numéro et son label dans le source, suivis de la liste des paramètres à passer à la routine et enfin de sa syntaxe Basic.

## Appeler une fonction depuis un programme en langage machine

Supposons que vous vouliez afficher un point en DHGR dans la couleur courante en X,Y=516,90 soit, en

hexadécimal, X=$204 et Y=$5A.

La fonction pour ce faire s'appelle DHPLOT0. Elle porte le numéro 9.

Son adresse est $4000+n*3=$401B.

Il est précisé, dans la description de l'instruction :

Parm : X = xl, Y = xh, A = y

xl représente l'octet de poids faible de x ; xh représente l'octet de poids fort dex.

Le programme appelant sera donc :

```
LDA £90          ou   LDA £$5A
LDX £<516        ou   LDX £$04
LDY £>516        ou   LDY £$02
JSR $4000+27     ou   JSR $401B
```

# Fonctions de "Double graphics"

Notes :

- les couleurs vont de 0 à 15 ;
- les abscisses DGR sont comprises entre 0 et 79 bornes incluses, et les ordonnées entre 0 et 47 bornes incluses ;
- en DHGR, ces 'gammes' sont 0 à 559 et 0 à 191 ;
- pour éviter toute confusion, les registres du 6502 sont désignés par des majuscules et les paramètres par des minuscules ;
- xh représente l'octet de poids fort de x, et xl l'octet de poids faible ;
- il est impératif de se mettre en 80 colonnes avant toute utilisation du graphisme.

## 00 INITDHGR

**Parm : -**
**Basic : CALL 16384**

Initialise l'amper-interpreteur. Automatiquement exécuté lors d'un BRUN DHGR ou d'un - DHGR.

## 01 FILLSCR

**Parm : A = couleur**
**Basic : & GR couleur**

Initialise le mode double-basse résolution, réduit la fenêtre texte et remplit l'écran avec la couleur indiquée. Exemple : l'équivalent du "GR" de l'Applesoft est "& GR 0".

## 02 PLOT

**Parm : Y = x, A = y**
**Basic : & PLOT x,y**

Affiche un point basse-résolution en x,y dans la couleur courante.

## 03 GRSCRN

**Parm : Y = x, A = y**
**Basic : & SCRN GR x,y,var**

Retourne la couleur du point aux coordonnées indiquées. En machine, la couleur est retournée dans A, et la retenue (C) est mise à 1 si cette couleur est différente de zéro (autrement dit, si le point est 'allumé'), à 0 sinon.

En Basic, la couleur est stockée dans la variable "var". Si cette dernière n'existe pas, elle est créée, comme dans toute affectation, tout comme le ferait l'Applesoft.

## 04 DOHLINE

**Parm : Y = x1, X = x2, A = y**
**Basic : & HLIN x1,x2 AT y**

Trace une ligne basse-résolution horizontale à l'ordonnée y, allant des abscisses x1 à x2, dans la couleur courante.

## 05 DOVLINE

**Parm : A = y1, X = y2, Y = x**
**Basic : & VLIN y1,y2 at x**

Trace une ligne basse-résolution verticale à l'abscisse x, de y1 à y2.

## 06 SETCOL

**Parm : A = couleur**
**Basic : & HCOLOR= couleur**

Spécifie la couleur à utiliser dans les tracés basse-résolution.

## 07 DRAWCHR

**Parm : X = n, Y = x, A = y**
**Basic : & DRAW n AT x,y**

Dessine en basse-résolution le caractère graphique numéro n à partir (coin supérieur gauche de la matrice du caractère) du point de coordonnées x,y, dans la couleur courante.

Si certains points se situent en dehors de l'écran, ils ne sont pas tracés. Il est donc possible de 'déborder' de l'écran.

## 08 INITHGR

**Parm : -**
**Basic : & HGR**

Initialise le mode double-haute résolution avec 4 lignes de texte en bas de l'écran. Noircit l'écran.

## 09 INITHGR2

**Parm : -**
**Basic : & HGR2**

Même fonction que INITHGR (cf ci-dessus), mais sur tout l'écran (pas de fenêtre texte en bas). Notez que cette instruction ne commute pas une seconde page écran, mais la MEME page graphique, non mixte.

## 10 DHPLOT0

**Parm : X = xl, Y = xh, A = y**
**Basic : & HPLOT x,y**

Affiche un point double-haute résolution aux coordonnées spécifiées, dans la couleur courante. Déplace le curseur graphique en x,y.

## 11 HGLIN

**Parm : A = xl, X = xh, Y = y**
**Basic : & HPLOT TO x,y**

Trace une ligne double-haute résolution depuis la position courante du curseur graphique jusqu'aux coordonnées x,y spécifiées.

Note : comme pour l'Applesoft, il est possible de chaîner les tracés. Par exemple :

```
&HPLOT 0,0 TO 559,0 TO 559,191
    TO 0,191 TO 0,0
```

trace un cadre rectangulaire marquant les limites de l'écran.

## 12 HGRSCRN

**Parm : X = xl, Y = xh, A = y**
**Basic : & HGR SCRN x,y,var**

Renvoie 1 si le point aux coordonnées x,y est allumé, 0 sinon. En machine, le résultat est renvoyé dans A et dans la retenue (C). En Basic, le résultat est dans "var". Notez que ce procédé permet de tester directement le résultat par BCS ou BCC en machine, et par "IF var THEN allumé"... en Basic.

## 13 SETHCOL

**Parm : A = couleur**
**Basic : & HCOLOR= couleur**

Spécifie la couleur double-haute résolution à utiliser.

## 13 SETHCOL

**Parm : A = $FF**
**Basic :**
**& HCOLOR= INVERSE**

Les tracés double-haute résolution se feront en négatif par rapport au fond. Par exemple, lorsqu'un point est tracé sur un fond blanc, il apparaît en noir (pensez à XDRAW). Ainsi, deux tracés successifs s'annulent (le négatif d'un négatif est un positif). En Basic, l'ancienne couleur est restaurée par & HCOLOR= NORMAL et en machine par LSR $305.

## 14 FILLZ

**Parm : A = n0, X = n1**
**Basic : & CLEAR n0,n1**

Remplit la fenêtre en utilisant l'octet n0 pour les lignes paires, et n1 pour les lignes impaires.

Attention : les 8 bits sont utilisés, et non 7 comme il est courant de voir. Ces 8 bits sont décalés sur 7 bits afin d'être tous affichés. Autrement dit, comme un pixel en couleur est codé sur 4 bits, il est possible de remplir la fenêtre avec des groupes de 4 points ayant chacun 16 couleurs possibles. Pour calculer l'octet correspondant à une paire de points de couleur c1 et c2, la formule est n0 (ou n1) = c1 * 16 + c2. Le point de couleur c1 sera à gauche. Par exemple, pour remplir la fenêtre avec une alternance de points blancs (couleur 15) et noirs (couleur 0), décalés de ligne en ligne :

```
& CLEAR 15,240
```

## 14 FILLZ

**Parm : $306 = $80**
**Basic : & CLEAR INVERSE**

Inverse la fenêtre (par EOR £$7F).

## xx SETWND

**(pas de vecteur)**

**Basic :**
**& HOME mg,md,mh,mb**

Définit la fenêtre. Les marges gauche et droite (mg et md) sont comprises entre 0 et 79, les marges haut et bas (mh,mb) entre 0 et 23. En machine, l'équivalent est obtenu en utilisant les adresses $20 à $23.

Dans le cas de FILLZ (& CLEAR), la fenêtre texte est convertie de façon à ce que la fenêtre DHGR utilisée couvre exactement la même surface au même endroit qu'en mode texte.

## xx DEFPAT

**(pas de vecteur)**

**Basic : & DEF n;n1,n2,... ,n8**

Redéfinit le caractère graphique n avec les 8 octets n1,... ,n8 (de haut en bas). Chaque caractère a une matrice hxv de 7 x 8 pixels. Chaque bit à 1 représente un point allumé, à 0 un vide.

## 15 HCOUT

**Parm : A = n, X = x, Y = y**
**Basic : & PRINT x,y,CHR$(n)**

Affiche le caractère graphique n en 7*x,y (0<x<79, car les caractères ne sont affichables que sur un octet entier (l'abscisse DHGR correspondante est 7*x), mais en revanche 0<y<191). Les codes 0 à 31 ne sont pas interprétés comme des codes de contrôle, mais comme des CARACTERES graphiques. Il n'y a donc pas de retour chariot, ni de saut de ligne.

Lorsque Y>184, il n'y a plus assez d'espace pour afficher le caractère dans toute sa hauteur : seule la partie à l'intérieur de l'écran est affichée.

De plus, il n'y a pas d'affichage texte simultané : cette routine et les suivante ne modifient que la page graphique DHGR.

Dans la mesure où j'ai voulu permettre l'utilisation de polices de 256 caractères, le mode inverse est calqué sur celui du Basic : il suffit donc d'utiliser les instructions INVERSE/NORMAL de l'Applesoft pour switcher le mode d'inversion du texte. Le code du caractère ne détermine donc pas son mode d'affichage.

## 16 STROUT

**Parm :**
**$5E = adr, X = x, Y = y**
**Basic : & PRINT x,y,exp**

La routine pour l'assembleur affiche aux coordonnées x et y (mêmes contraintes que pour HCOUT) la chaîne dont l'adresse est en $5E (INDEX). La fin de la chaîne est marquée par l'octet 00. Le caractère graphique 00 ne peut donc être affiché par STROUT. La longueur maximale de la chaîne est de 255 caractères.

En Basic, exp est affichée en x,y. exp peut être une chaîne, un nombre ou une expression. Il est possible d'écrire

```
& PRINT x,y,exp1;exp2;...
```

auquel cas exp2 sera affichée immédiatement à la suite de exp1.

Arrivé à la marge droite, un retour à la ligne est effectué. Les marges haut et bas ne sont pas prises en compte, puisqu'il n'y a pas de 'scrolling' (trop lent en DHGR). Le système adopté est celui de l'affichage par 'page' : arrivé en bas de l'écran (y>191), on continue en haut (y=0).

## 17 FNDADR

**Parm : A = n**
**Pas d'équivalent Basic (lié à DEFPAT)**

Calcule l'adresse des 8 octets de la matrice du caractère. Le résultat est dans X (adrl) et Y (adrh), ainsi que dans INDCHR ($CE). Cette routine sert à accéder aux caractères graphiques par exemple pour les modifier.

## 18 SETMOD

**Parm : A = m**
**Basic : & LET m**

Sélectionne le mode d'affichage des caractères en DHGR. Avant d'être transféré dans la mémoire vidéo, chaque octet de la matrice des caractères subit une opération logique avec l'octet de l'écran correspondant.

Ces opérations sont les suivantes :

- m = 0 : rien (affichage normal), le fond est recouvert ;
- m = 1 : OU INCLUSIF (ORA), les caractères sont superposés au décor ;
- m = 2 : OU EXCLUSIF (EOR), chaque pixel du caractère est le complémentaire du fond. À un point noir correspond un point blanc et vice-versa. C'est le principe de la pseudo-couleur 'inverse' ;
- m = 3 : ET (AND), seuls restent allumés les pixels qui sont à 1 à la fois dans le caractère et dans le fond ;
- m = 4 : ZERO, les caractères ne sont pas affichés.

## 19-22 USER1-4

**Parm : ...**
**Basic: & USR n,...**

Saute à la routine utilisateur numéro n, qui prend alors en main l'évaluation et le contrôle des paramètres. Les 4

vecteurs pointent normalement sur INITDHGR ($4000). En Basic, la virgule après n étant facultative, elle n'est pas sautée automatiquement.

## Source 'DHGR.S'

**Assembleur Merlin-Pro**

```
*****************************
*      DOUBLE  GRAPHICS     *
*      ---------------      *
* (C) Pascal CANTOT, Juillet 85»
*      mai 87 et Juillet 87  *
*****************************

VALTYP  =   $11
CNTR    =   $18
TEMP    =   $19
COUNTH  =   $1D
WNDLFT  =   $20
WNDWDTH =   $21
WNDTOP  =   $22
WNDBTM  =   $23
HBASR   =   $26
GBASE   =   $26
H2      =   $2C
V2      =   $2D
TOKEN   =   $2E
MASK    =   $2E
COLOR   =   $30
HBASR   =   $30
INVFLG  =   $32
LINNUM  =   $50
TEMPPT  =   $52
INDEX   =   $5E
DSCTMP  =   $9D
CHRGET  =   $B1
CHRGOT  =   $B7
INDCHR  =   $CE
DX      =   $D0
DY      =   $D2
CHRNT   =   $D3
EE      =   $D4
X0      =   $E0
Y0      =   $E2
WNDX    =   $E5
HPAG    =   $E6
ADRCTBL EQU $300
CARWDTH EQU $304
INVCOL  EQU $305
INVCOL2 EQU $306
PARFLAG EQU $307
AMPVECT =   $3F5
SET80COL =  $C00D
TXTCLR  =   $C050
FULL    =   $C052
MIXED   =   $C053
PAGE1   =   $C054
PAGE2   =   $C055
GHIRES  =   $C05E
IOUDIS  =   $C07E
ERROR   =   $D412
FRMNUM  =   $DD67
FRMEVL  =   $DD7B
CZKCOM  =   $DEBE
SYNCHR  =   $DEC0
PTRGET  =   $DFE3
ISLETC  =   $E07D
IQERR   =   $E199
GIVAYF  =   $E2F2
STRLIT  =   $E3E7
PREFAC  =   $E600
GETBYT  =   $E6F8
COMBYTE =   $E74C
GETADR  =   $E752
MOVMF   =   $EB2B
FOUT    =   $ED34
HGR     =   $F362
UPDOWN  =   $F4D3
MSKCTBL =   $F5B2
GBCALC  =   $F847
SETCOL  =   $F864
BASCALC =   $FBC1
        ORG $4000
* Table de vecteurs pour l'appel des routines
* depuis un programme machine exterieur:
        JMP INITDHGR   ;Initialise Ampersand
        JMP FILLSCR    ;Init DGR et
                        remplit ecran
        JMP PLOT       ;Affiche un point
                        basse-resolution
        JMP GRSCRN     ;Retourne la couleur
                        du point b.r.
        JMP DOHLINE    ;Trace ligne horizon
        JMP DOVLINE    ;Trace ligne verticale
        JMP SETCOL     ;Selection couleur DGR
        JMP DRAWCHR    ;Dessine caractere DGR
        JMP INITHGR    ;Mode DHGR mixte
        JMP INITHGR2   ;Mode DHGR plein ecran
        JMP DHPLOT0    ;Affiche un point DHGR
        JMP HGLIN      ;Trace une ligne DHGR
        JMP XGRSCRN    ;Teste le point
                        haute-resolution
        JMP SETHCOL    ;Selectionne couleur
        JMP FILL2      ;Remplit fenetre DHGR
        JMP HCOUT      ;Affiche un caractere
                        graphique
        JMP HSTROUT    ;Affiche chaine DHGR
        JMP FNDADR     ;Calcule adr. carect.
        JMP SETMOD     ;Selectionne le mode
                        d'affichage DHGR
USERS   JMP SNTXERR    ;Vecteurs pour ajouter
        JMP SNTXERR    ;nouvelles fonctions
        JMP SNTXERR
        JMP SNTXERR
*****************************
INITDHGR LDY £2        ;Revectorise l'&
AMPERLUP LDA JMPDHGR,Y
         STA AMPVECT,Y
         DEY
         BPL AMPERLUP
         RTS
JMPDHGR  JMP DHGR
NOTFND   LDA TOKEN     ;(TXTPTR) -> A
JMPERR   JMP SNTXERR   ;Peut etre change pour
                        utiliser un autre
*                       amper-interpreteur!
*****************************
DHGR     JSR CHRGOT    ;Saisit la commande
         STA TOKEN
         LDX £0
DHGR2    LDA CMDTBL,X  ;Verifie la commande
         BEQ NOTFND    ;Si fin de table des
                        commandes
         CMP TOKEN
         BEQ DHGR3     ;Si commande trouvee
         INX
         BPL DHGR2     ;Toujours pris
DHGR3    TXA
         ASL           ;lorsque commande est
         TAX           ;trouvee on
         LDA ADRTBL,X  ;recupere l'adresse
         STA GOCMD+1   ;de la routine
         LDA ADRTBL+1,X
         STA GOCMD+2
         JSR CHRGET
GOCMD    JMP RTS1      ;Adresse "bidon"
* Table codes ("tokens") des commandes AppleSoft
CMDTBL   HEX 88        ;GR
         HEX 8D        ;PLOT
         HEX A0        ;COLOR=
         HEX 8E        ;HLIN
         HEX 8F        ;VLIN
         HEX 93        ;HPLOT
         HEX BA        ;PRINT
         HEX 92        ;HCOLOR=
         HEX B6        ;GET
         HEX B8        ;DEF
         HEX 91        ;HCH
         HEX 90        ;HGR2
         HEX 94        ;DRAW
         HEX 97        ;HOME
         HEX BD        ;CLEAR
         HEX AA        ;LET
         HEX D5        ;USR
         BRK           ;Fin de la table
* Table des adresses des routines
ADRTBL   DA  XGR
         DA  XPLOT
         DA  XCOLOR
         DA  XHLIN
         DA  XVLIN
         DA  DHPLOT
         DA  HPRINT
         DA  XCOLOR
         DA  XSCRN
         DA  DEFPAT
         DA  INITHGR
         DA  INITHGR2
         DA  DRAW
         DA  SETWND
         DA  FILL
         DA  XHOME
         DA  XUSER
*****************************
XUSER    JSR GETBYT     ;numero vecteur (0-3)
         CPX £4
         BCS GOIQERR
         STX CNTR
         TXA
         ASL            ;Calcule USERS + n*3
         ADC CNTR       ;forcement C=0 car n<4
         ADC £<USERS    ;de meme C=0 ici aussi
         STA USRJMP+1   ;TOUS vecteurs USERS
USRJMP   JMP USERS      ;sur la meme page!
SETWND   JSR GETBYT     ;Redefinit la fenetre
                         texte
         CPX £80        ;(utile pour FILL)
         BCS GOIQERR
         STX TEMP       ;Saisit dans l'ordre:
         JSR COMBYTE    ;XMIN,XMAX,YMIN,YMAX
         CPX £80        ;avec X entre 0 et 79
         BCS GOIQERR    ;et Y entre 0 et 23
         CPX TEMP
         BCC GOIQERR    ;Erreur si XMAX < XMIN
         TXA
         SBC
         SBC TEMP       ;Effectue XMAX-XMIN+1
         STA CNTR       ;-> largeur fenetre
         JSR COMBYTE
         STX CNTR2
         CPX £24
         BCS GOIQERR
         JSR COMBYTE
         CPX £24
         BCS GOIQERR
         CPX CNTR2      ;Erreur si YMAX < YMIN
         BCC GOIQERR
         STX WNDBTM
         LDA CNTR2      ;En cas d'erreur
         STA WNDTOP     ;la fenetre N'EST PAS
                         modifiee
         LDA CNTR
         STA WNDWDTH
         LDA TEMP
         STA WNDLFT
         INC WNDWDTH
         INC WNDBTM
         JMP BASCALC
GOIQERR  JMP IQERR      ;"ILLEGAL Q. ERROR"
XGR      LDA COLOR
         STA TEMP2
         JSR XCOLOR
         LDA £20
         STA WNDTOP
         JMP FILLSCR2
XSCRN    JSR CHRGOT
         PHA
         JSR CHRGET
         PLA
         CMP £$88       ;GR?
         BEQ XGRSCRN
         CMP £$91       ;HGR?
         BEQ XHRSCRN
         JMP SNTXERR
XGRSCRN  JSR GETGRXY
         JSR GRSCRN
XSCRN2   PHA
         JSR CHKCOM
         PLA
         TAY
         LDA £0         ;Resultat -> Variable
         JMP AYVAR      ;A,Y -> variable
XGRSCRN  JSR HFNS
         JSR HGRSCRN
         SEC
         BCS XSCRN2     ;Toujours pris
XPLOT    JSR GETGRXY    ;Saisit coordonnees
         JMP PLOT       ;et affiche le point
XHLIN    JSR GETGRX     ;Saisit les abscisses
         STX TEMP+1
         JSR CZKCOM     ;Controle la virgule
         JSR GETGRX
         STX TEMP
         JSR CHKAT      ;Controle le "AT"
         JSR GETGRY     ;Saisit l'ordonnee
         TXA
         LDX TEMP
         LDY TEMP+1
         JMP DOHLINE    ;et trace la ligne
XVLIN    JSR GETGRY
         STX TEMP+1
         JSR CZKCOM
         JSR GETGRY
         STX TEMP
         JSR CHKAT
         JSR GETGRX
         TXA
         TAY
         LDA TEMP+1
         LDX TEMP
         JMP DOVLINE
XCOLOR   JSR GETBYT     ;Expression -> X
         CPX £16        ;Err. si couleur >=16
         BCS ILLQTY
         TXA
         JMP SETCOL
CHKAT    JSR CHRGOT
         CMP £$C5       ;"AT"
         BNE SNTXERR
         JMP CHRGET
SNTXERR  LDX £16
         JMP ERROR
CNTR2    DS  1
TEMP2    DS  1
```

**LES ROUTINES 'DHGR' FONCTIONNENT SOUS PRODOS ET SOUS DOS 3.3 SUR LES Apple //e 128Ko, //c et IIGS.**

```
*******************************
ILLQTY   JMP   IQERR
GETGRXY  JSR   GETGRX
         STX   TEMP
         JSR   CHKCOM      ;Saute la virgule
         JSR   GETGRY
         TXA
         LDY   TEMP
         RTS
GETGRX   JSR   GETBYT
         CPX   £80
         BCS   ILLQTY      ;Erreur si X>79
         RTS
GETGRY   JSR   GETBYT
         CPX   £48         ;Erreur si Y>=48
         BCS   ILLQTY
         RTS
AYVAR    JSR   GIVAYF      ;A,Y -> Fac
         JSR   PTRGET      ;Cherche variab en RAM
         TAX               ;(la cree si elle
                            n'existe pas)
         JMP   MOVMF       ;puis Fac -> Variable
*******************************
FILLSCR  LDA   COLOR
         STA   TEMP2
         JSR   SETCOL
FILLSCR2 STA   TXTCLR
         STA   SET80COL    ;Initialise mode DGR
         STA   IOUDIS
         STA   DHIRES
         STA   MIXED
         STA   $C056
         STA   $C000
         LDA   COLOR
         STA   TEMP
         JSR   EVCOL
         LDA   £0
         STA   CNTR
FILSCR2  LDA   CNTR
         JSR   GBCALC
         LDA   COLOR
         LDY   £39
FILSCR3  STA   $C005
         LDA   COLOR
         STA   (GBASE),Y
         STA   $C004
         LDA   TEMP
         STA   (GBASE),Y
         DEY
         BPL   FILSCR3
         INC   CNTR
         LDA   CNTR
         CMP   £20
         BCC   FILSCR2
         LDA   TEMP2
         STA   COLOR
         RTS
GRSCRN   PHA
         LDA   £$91        ;Opcode de LDA (xx),Y
         STA   IOBYT2      ;(empeche PLOT
                            d'afficher le point)
         PLA
         JSR   PLOT        ;Utilise PLOT pour
                            recuperer
         LDA   £$B1
         STA   IOBYT2
         LDA   MASK        ;l'octet de l'ecran
                            correspondant
         EOR   £$FF        ;(PLOT le range dans
                            TEMP2)
         STA   TEMP
         LDA   TEMP2
         AND   MASK
         CMP   £$10
         BCC   GRSCRN3     ;Si 0000xxxx
         LSR               ;Si xxxx0000, decale
                            4 fois a droite
         LSR
         LSR
         LSR
GRSCRN2  BIT   PARFLAG     ;Colonne paire?
         BPL   GRSCRN5     ;Non->Couleur correcte
         LDY   £0          ;Oui->Conversion
         STA   TEMP
         ASL
         ASL
         ASL
         ASL
         ORA   TEMP
GRSCRN3  CMP   COLTBL,Y
         BEQ   GRSCRN4
         INY
         BPL   GRSCRN3     ;Toujours pris
GRSCRN4  TYA
GRSCRN5  AND   £$0F
         CMP   £1          ;Resultat (couleur
                            0-15) dans A
         RTS               ;avec C=1 si non noir
PLOT     LSR
         PHP
         LSR   PARFLAG
         JSR   GBCALC
         PLP
         LDA   COLOR
         PHA
         LDA   £$0F
         BCC   RTMASK
         LDA   £$F0
RTMASK   STA   MASK
PLOT1    TYA               ;PLOT a la derniere
                            ordonnee (X->Y)
         PHA
         STA   $C001       ;Inhibe le commutateur
                            RAMWRT
         STA   PAGE1
         LSR               ;Si pair, mev aux.
         TAY
         BCS   PLOT2
         SEC
         EOR   PARFLAG
         STA   PAGE2
         TYA
         PHA
         JSR   EVCOL
         PLA
         TAY
PLOT2    LDA   (GBASE),Y
         STA   TEMP2
         EOR   COLOR
         AND   MASK
         EOR   (GBASE),Y
IOBYT2   STA   (GBASE),Y
         STA   PAGE1
         PLA
         TAY
         PLA
         STA   COLOR
         CLC
         RTS
DOHLINE  STX   H2
HLINE    JSR   PLOT        ;Trace ligne horizon.
HLINE1   CPY   H2          ;X1 a X2 a ordonnee Y
         BCS   RTS1        ;Y>A,X1>Y,X2>R2 ($2C)
         INY
         JSR   PLOT1
         BCC   HLINE1
DOVLINE  STX   V2
         SEC
         BCS   VLINE       ;Toujours pris
VLINE2   ADC   £1
VLINE    PHA               ;Trace ligne verticale
         JSR   PLOT        ;X1 a Y2 a abscisse X
         PLA               ;Y1>A,X>Y,Y2>V2 ($2D)
         CMP   V2
         BCC   VLINE2
RTS1     RTS
EVCOL    LDA   COLOR
         AND   £$0F
         TAY
         LDA   COLTBL,Y
         STA   COLOR
         RTS
COLTBL   HEX   0088119922AA33BB
         HEX   44CC55DD66EE77FF
DRAW     JSR   GETBYT      ;Dessine un caractere
         STX   CNTR        ;basse-resolution
         JSR   CHKAT
         JSR   GETGRXY
         LDX   CNTR
DRAWCHR  STX   CHAR        ;Y->A, X->Y, Char->X
         STA   YPOS
         STY   XPOS
         STY   LINNUM
         TXA
         JSR   FNDADR      ;Calcule adr. caract.
         JSR   SETIMSK     ;Calcule masque
                            d'inversion
         LDA   £8          ;Matrice du caractere
         STA   CNTR        ;comprend 8 octets
                            (8x8 points)
ADRCHR2  EQU   *+1         ;(et non 7x8 comme en
                            DRGR!)
DRCHR1   LDA   $2000       ;Adresse (arbitraire)
                            de l'octet
         EOR   INVMASK     ;Inverse la matrice
                            si mode INVERSE
         LDX   £8          ;Tester les 8 bits
         STX   CNTR2
         LDY   LINNUM
         STY   XPOS
DRCHR2   LSR               ;Decale a droite.
         BCC   DRCHR4      ;si C=0, pas de point
         PHA               ;si C=1, un point
         LDA   YPOS
         LDY   XPOS
         CMP   £48         ;Sortie de l'ecran
         BCS   DRCHR3      ;alors n'affiche pas
         CPY   £80         ;le point
         BCS   DRCHR3
         JSR   PLOT
DRCHR3   PLA
DRCHR4   INC   XPOS
         DEC   CNTR2
         BNE   DRCHR3
         INC   YPOS
         INC   ADRCHR2
         BNE   DRCHR3
         INC   ADRCHR2+1
DRCHR3   DEC   CNTR
         BNE   DRCHR1
         RTS
******* HIRES ROUTINES *******
HIRES2   SEC
         REX   24
HIRES    CLC
         STA   DHIRES
         STA   TXTCLR      ;GR
         STA   $C057       ;Puis HGR
         STA   MIXED
         BCC   HIREND
         STA   FULL        ;Full graphics
HIREND   RTS
HCOLOR   JSR   CHRGOT
         CMP   £$9E        ;"Couleur" inverse?
         BNE   HCOLOR2     ;Si INVERSE, alors C=1
HCOLOR20 ROR   INVCOL      ;1 -> Bit7 de INVCOL
         JMP   CHRGET
HCOLOR2  CMP   £$9D        ;Couleur normale ?
         BNE   HCOLOR3
         CLC               ;C=0
         BCC   HCOLOR20    ;Toujours pris
HCOLOR3  JSR   GETBYT
         TXA
SETHCOL  CMP   £$FF        ;Couleur "inverse" ?
         BEQ   HCOLOR20    ;Si pris, alors C=1
         LSR   INVCOL
         STA   HCOL
         ASL
         ASL
         ASL
         ASL
         ORA   HCOL
         STA   HCOL
         RTS
HCOL     HEX   FF          ;Init : Blanc
DEFPAT   JSR   GETBYT      ;Code ASCII du caract.
         TXA
         JSR   FNDADR
         LDA   £';'
         JSR   SYNCHR
         LDA   £7
         STA   CNTR
         BNE   DEFPAT2
DEFPAT1  JSR   CHRCOM
DEFPAT2  JSR   CNTRYT
         LDY   CNTR
         TXA
         STA   (INDCHR),Y
         DEC   CNTR
         BPL   DEFPAT1
         RTS
INITHGR  STA   $C000
         STA   $C005
         JSR   HGR         ;Initialise HGR1
         STA   $C004
         JSR   HGR
         JMP   HIRES
INITHGR2 JSR   INITHGR
         JMP   HIRES2
FILL     LSR   INVCOL2
         JSR   CHRGOT
         CMP   £$9E        ;Inversion demandee?
         BNE   FILL1       ;Non->Saisie 2 octets
         ROR   INVCOL2     ;Sinon, C=1 -> Bit7 de
                            INVCOL2
         JSR   CHRGET
         JMP   FILL2
FILL1    JSR   GETBYT      ;Saisit paire d'octets
         TXA               ;Le premier (A) pour
                            lignes paires
         PHA               ;le second (X) pour
                            lignes impaires
         JSR   COMBYTE
         PLA
FILL2    STX   LINNUM+1
         STA   LINNUM
         LDA   WNDBTM      ;Convertit fenetre txt
         ASL               ;en coord.exploitables
         ASL
         ASL
         STA   CNTR2
         LDA   WNDLFT
         CLC
         ADC   WNDWDTH
         STA   WNDRGHT
         LDA   WNDTOP
         ASL
         ASL
         ASL
         STA   LINE
         STA   DHIRES      ;S'assure DHIRES ON,
FILL2    LDA   LINE
         JSR   HPOSN       ;Calc. adr. ligne
         LDA   WNDLFT      ;Commence a gauche
```

```
        STA   CNTR
        LDY   £0
        LDA   LINE
        LSR                 ;Teste parite ligne
        BCC   FILEVEN       ;Si paire
        INY                 ;Impaire, oct. suivant
FILEVEN LDA   LINNUM,Y      ;Prend oct. remplissag
        STX   TEMP
        STX   $C001
FILL3   STA   PAGE1
        LDA   CNTR
        LSR                 ;Teste parite colonne
        BCS   FILL4         ;Si MEV principal
        STA   PAGE2         ;Si MEV aux.
FILL4   TAY
        BIT   INVCOL2       ;Inversion?
        BMI   INVFILL
        LDA   TEMP
        STA   (HBASE),Y
        ROL                 ;rotation de l'octet
                             vers la gauche:
        ROL   TEMP          ;On decale, bit7->C,
                             puis encore:
                             bit7->bit0
FILL5   INC   CNTR          ;Octet suivant
        LDA   CNTR
        CMP   WNDRGHT       ;Jusqu'a marge droite
        BCC   FILL3
        INC   LINE          ;Ligne suivante
        LDA   LINE
        CMP   CNTR2         ;Jusqu'a marge du bas
        BCC   FILL2
        STA   PAGE1
        RTS
INVFILL LDA   (HBASE),Y
        EOR   £$7F          ;N'inverse pas le bit7
        STA   (HBASE),Y
        SEC
        BCS   FILL5         ;Toujours pris
HCOL1   DS    1
WNDRGHT DS    1
LINE    DS    1
HGRSCRN PHA
        LDA   £$B1          ;Opcode de LDA (xx),Y
        STA   IOBYT1
        PLA
        JSR   DHPLOT0       ;Modifie DHPLOT1
                             pour recuperer
        LDX   £$91          ;Opcode de STA (xx),Y
        STX   IOBYT1
        AND   HMASK
        BEQ   HGRSCRN2      ;Si zero, point etein
        LDA   £1            ;Sinon, allume
        SEC
        HEX   24            ;Truc! (Opc de BIT zp)
HGRSCRN2 CLC                ;(0 ou 1) dans A et C
        RTS
DHPLOT  JSR   CHRGOT
        CMP   £$C1          ;"TO"?
        BEQ   HP3           ;Oui-> ligne
        JSR   HFNS          ;Prend coordonnees
                             point de depart
HP1     JSR   DHPLOT0       ;Affiche le point
HP2     JSR   CHRGOT
        BEQ   HPLOTEND
        CMP   £$C1          ;Ligne?
        BNE   HPLOTEND      ;Non-> Fin
HP3     JSR   CHRGET
        JSR   HFNS          ;Prend les coordonnees
                             de destination
        STX   X1
        STY   X1+1
        STX   Y1
        STY   DSCTMP
        TAY
        TXA
        LDA   DSCTMP
        JSR   HGLIN         ;Plot line
        LDX   X1
        LDY   X1+1
        LDA   Y1
        JMP   HP1
X1      DS    2
Y1      DS    1
HPLOTEND RTS
HGLIN   PHA                 ;En entree:
        SEC                 ; XL: A-reg
        SBC   X0            ; XH: X-reg
        PHA                 ; Y:  Y-reg
        TXA
        SBC   X0+1
        STA   QDRNT         ;Calcule ABS(X-X0)
        BCS   HLIN2         ; dans (DX,H)
        PLA
        EOR   £$FF          ;dir X -> bit de signe
        ADC   £1            ; de qdrnt.
        PHA                 ; 0 = droite (DX pos)
        LDA   £0            ; 1 = gauche (DX neg)
        SBC   QDRNT
HLIN2   STA   DX+1
        STA   EE+1          ;Initialise EE,H a
                             ABS(X-X0)
        PLA
        STA   DX
        STA   EE
        PLA
        STA   X0
        STX   X0+1
        TYA
        CLC
        SBC   Y0            ;Calcule -ABS(Y-Y0)-1
                             dans DY
        BCC   HLIN3
        EOR   £$FF
        ADC   £$FE
HLIN3   STA   DY            ;Rotation Y dir dans
        STY   Y0            ;le bit de signe de
                             qdrnt
        ROR   QDRNT         ;(0 = haut, 1 = bas)
        SEC
        SBC   DX            ;Initialise COUNTL,H
        TAX                 ; a -(DELTX,DELTY+1)
        LDA   £$FF
        SBC   DX+1
        STA   COUNTH
        LDY   HNDX          ;Index horizontal.
        BCS   MOVEX2        ;Toujours pris.
MOVEX   ASL                 ;Mvt dans X-dir.
                             Utilise
        JSR   LFTRT         ; bit6 de QDRNT bit6
                             pour g/dr.
        SEC
MOVEX2  LDA   EE            ;Suppose C=1
        ADC   DY            ;(EE,H)-DELTY > (EE,H)
        STA   EE            ;Note: DY = (-DELTY)-1
        LDA   EE+1          ;C=0 si (EE,H) negatif
        SBC   £0
HCOUNT  STA   EE+1
        PHP
        STY   HNDX
        JSR   DHPLOT1       ;Affiche point
        LDY   HNDX
        PLP
        INX                 ;On a trace
                             (DELTX+DELTY) points?
        BNE   HLIN4
        INC   COUNTH
        BEQ   HLINEND       ;Si oui, termine.
HLIN4   LDA   QDRNT         ;Tester la direction.
        BCS   MOVEX         ;Si C=1, affiche point
                             (EE,H)
        JSR   UPDOWN        ;C=0, neg, deplacement
        CLC
        LDA   EE            ;(EE,H)+DELTX > (EE,H)
        ADC   DX
        STA   EE
        LDA   EE+1
        ADC   DX+1
        BVC   HCOUNT        ;Toujours pris.
HLINEND RTS
LFTRT   BPL   RIGHT         ;Signe : LEFT/RIGHT
LEFT    LDA   HMASK
        LSR
        BCS   LEFT1
        EOR   £$C0
        STA   HMASK
        RTS
LEFT1   DEY
        BPL   LEFT2
        LDY   £79
LEFT2   LDA   £$C0
        BNE   NEWNDX        ;Toujours pris
RIGHT   LDA   HMASK         ;pour LEFT mais pas
                             pour RIGHT
        ASL
        EOR   £$80
        BPL   LR1
        STA   HMASK
        RTS
LR1     LDA   £$81
        INY
        CPY   £80           ;40 pour la ROM!
        BCC   NEWNDX
        LDY   £0
NEWNDX  STY   HNDX
        STA   HMASK
        PHP
        JSR   HPOSN6
        PLP
        RTS
DHPLOT0 STA   $C000
        JSR   HPOSN
DHPLOT1 LSR   HNDX          ;Bit 0 -> Carry
        LDA   HNDX
        STA   $C001
        STA   PAGE1
        BCS   MAINRAM
        STA   PAGR2
MAINRAM LDY   HNDX
        BIT   INVCOL        ;"Couleur" inverse?
        BPL   DHPLOT2
        LDA   (HBASE),Y
        EOR   HBASE
        JMP   DHPLOT3
DHPLOT2 LDA   HCOL1
        EOR   (HBASE),Y
        AND   HMASK
        EOR   (HBASE),Y
DHPLOT3 AND   £$7F
IOBYT1  STA   (HBASE),Y
        STA   PAGE1
        STA   $C000
        ROL   HNDX
        RTS
HFNS    JSR   FRMNUM
        JSR   GETADR        ;Evalue expression X
        LDY   LINNUM+1
        LDX   LINNUM        ;Controle validite
        CPY   £>560         ;(doit etre < 560)
        BCC   HFNS1
        BNE   ILQTY         ;sinon, erreur
        CPX   £<560
        BCS   ILQTY
HFNS1   JMP   $F8CD         ;On laisse l'Applesoft
*                          ;faire le reste
ILQTY   JMP   IQERR
HPOSN   STX   X0            ;Calcule adr. (HBASE)
        STY   X0+1          ;et msq (HMASK) pour
        CMP   Y0            ;point specifie:
        BEQ   HPOSN2        ;En entree: Y->A,
                             Xl->X, Xh->Y
        STA   Y0
        PHA
        AND   £$C0
        STA   HBASE
        LSR
        LSR
        ORA   HBASE
        STA   HBASE
        PLA
        STA   HBASE+1
        ASL
        ASL
        ASL
        ROL   HBASE+1
        ASL
        ROL   HBASE+1
        ASL
        ROR   HBASE
        LDA   HBASE+1
        AND   £$1F
        ORA   HPAG
        STA   HBASE+1
HPOSN2  TXA                 ;Divise X0 par 7
        CPY   £0            ;quotient = Numero de
                             l'octet (0-79)
        BEQ   HPOSN5        ;reste = index MMASK
        DEY                 ;X > 255
        BEQ   HPOSN2        ;Si X < 512
HPOSN1  LDY   £72
        ADC   £1
        BNE   HPOSN4        ;Toujours pris
HPOSN2  CMP   £$FB          ;Cas ou X>507
        BCC   HPOSN3
        CMP   £$FF
        BNE   HPA
        LDA   £$F9          ;Masque £0 si X=511
        LDY   £73
        BNE   SETMSK
HPA     ADC   £1            ;Note: C=0
        LDY   £72           ;Masque £(XL-$F9+1)
        BNE   SETMSK        ;si X entre 507 et 510
HPOSN3  LDY   £$23
        ADC   £5
HPOSN4  INY
HPOSN5  SBC   £7
        BCS   HPOSN4
SETMSK  STY   HNDX
        TAX                 ;X = index dans MSKTBL
        LDA   MSKTBL-$F9,X
        STA   HMASK
HPOSN6  TYA
        AND   £$0F
        STA   TEMP
        TAX
        LDA   HCOL
        BEQ   HPOSEND       ;Inutile de decaler
        CMP   £$FF          ;blanc et noir
        BEQ   HPOSEND
        CPX   £0
        BEQ   HPOSEND
HPOSN7  PHA                 ;Ajuste autres couleurs
        ROL                 ;qui sont sur 4 bits
        PLA                 ;a l'interieur 7 bits
        ROL                 ;chaque octet page HGR
        DEX
        BNE   HPOSN7
HPOSEND STA   HCOL1
        LDX   TEMP
        RTS
*****************************
XMODE   JSR   GETBYT        ;Gestion commande
        CPX   £5            ;de changement de mode
        BCC   SETMOD        ;d'affichage
        JMP   IQERR
SETMOD  LDA   OPCTBL,X      ;X = mode (0-4)
```

```
        STA   IOBYTE
        RTS
OPCTBL  DFB   $91,$11,$51,$31,$B1 ;Opcodes STA
                                   ORA,EOR,AND,LDA
HPRINT  JSR   GETBYT     ;Gestion cmde "PRINT"
        CPX   £80
        BCS   ILLQERR
        STX   OURCH
        JSR   CHKCOM
        JSR   GETBYT
        CPX   £192
        BCS   ILLQERR
        STX   OURCV
        JSR   CHKCOM
HPRINT2 JSR   CHRGOT
        BEQ   HPRTEND    ;Si zero, c'est fini!
        BCC   RDVAR      ;c'est un nombre
        JSR   ISLETC     ;car. alpha ?
        BCS   RDVAR      ;C'est une variable
        CMP   £$22
        BEQ   RDVAR      ;chaine entre ""
        CMP   £";"
        BEQ   HPRINT2
HPRTEND RTS
OURCH   DS    1
OURCV   DS    1
ILLQERR JMP   $E199
RDVAR   LDA   £0
        STA   TEMPPT
        JSR   FRMEVL     ;Evalue formule
        BIT   VALTYP
        BNE   STRNGOUT   ;C'est une chaine
        JSR   FOUT       ;C'est un nombre
        JSR   STROUT
        JMP   HPRINT2
STRNGOUT JSR  STROUT2
        JMP   HPRINT2
HSTROUT STX   OURCH
        STY   OURCV
        JMP   STROUT2
STROUT  JSR   STRLIT
STROUT2 JSR   FREFAC
        TAX
        BEQ   HPRTEND    ;Si chaine vide
STROUT2 LDY   £0
STRLUP  LDA   (INDEXU),Y
        BEQ   HPRTEND
        JSR   HOUTCHR
        DEX
        BEQ   HPRTEND
        INY
        BNE   STRLUP
        RTS
HOUTCHR STA   CHAR       ;Affiche un caractere
        TYA              ;avec deplacmt curseur
        PHA              ;(modifie pas A,X,Y)
        TXA
        PHA
        LDX   OURCH
        LDA   OURCV
        TAY
        LDA   CHAR
        JSR   HCOUT
        INC   OURCH
        BIT   CARWDTH
        BPL   OUTCHR2
        INC   OURCH
OUTCHR2 LDA   OURCH
        CMP   £80
        BCC   OUTCHR4
        LDA   £0
        STA   OURCH
        LDA   OURCH
        CLC
        ADC   £8
        CMP   £192       ;Arrive bas ecran ?
        BCC   OUTCHR3    ;on revient en ligne 0
        LDA   £0
OUTCHR3 STA   OURCV
OUTCHR4 PLA              ;Restaure A,X,Y
        TAX
        PLA
        TAY
        RTS
HCOUT   STA   CHAR       ;Affiche car. dont le
        STX   XPOS       ;code ascii est dans A
        STY   YPOS       ;aux coords (X+7,Y)
        STX   CNTR2
        LDA   YPOS
        LSR   PARITY
        LDY   £0
        LDA   CHAR
        JSR   SETIMSK
        LSR   CNTR2      ;Teste parite de CNTR,
                          alias XPOS
        ROR   PARITY     ;Parity=$80 si MEV
                          princ.
        LDA   CHAR
        JSR   FNDADR     ;Calc. adr. caractere
        LDX   £8
        STX   CNTR
HLUP    LDA   YPOS
        CMP   £192       ;On sort de l'ecran?
        BCS   HCONT
        JSR   HPOSN
        LDY   CNTR2
ADRCHR  =     *+1
        LDA   $2000      ;Valeur arbitraire
        EOR   INVMASK
        AND   £$7F       ;Supprime le bit7
        BIT   CARWDTH    ;Si -, -> dble largeur
        BMI   DBLCAR
        JSR   PUTBYTE
HCONT   INC   YPOS       ;Sortie de l'ecran,
        JSR   INCADR     ;cont. traiter autres
        DEC   CNTR       ;lignes afin conserver
        BNE   HLUP       ;une vitesse constante
                          d'affichage
        LDA   CHAR       ;Registres inchangés!
        LDX   XPOS
        LDY   YPOS
        RTS
DBLCAR  JSR   DSHIFT
        LDA   LINNUM
        JSR   PUTBYTE
        LDA   PARITY
        PHA
        EOR   £$80       ;Inverse la parite
        STA   PARITY
        BMI   DBLCAR2    ;Si -, meme adresse
        INY              ;sinon l'octet suivant
        CPY   £40        ;Sort-on de l'ecran?
        BCS   DBLCAR3
DBLCAR2 LDA   LINNUM+1
        JSR   PUTBYTE
DBLCAR3 PLA              ;restaure flag parite
        STA   PARITY
        JMP   HCONT
PUTBYTE STA   $C001      ;Transfere un octet
        STA   $C054      ;vers la page DHGR
        BIT   PARITY
        BMI   PUTMAIN
        STA   $C055
PUTMAIN ORA   £$80
IOBYTE  STA   (RBASE),Y
        STA   (HBASE),Y
        STA   $C054
        RTS
INCADR  INC   ADRCHR
        BNE   INCADR2
        INC   ADRCHR+1
INCADR2 RTS
CHAR    DS    1
XPOS    DS    1
YPOS    DS    1
INVMASK DS    1
PARITY  DS    1
SETIMSK LDX   £0         ;Inverse msg inversion
        BIT   INVFLG
        BMI   SIMSK2
        LDX   £$FF
SIMSK2  STX   INVMASK
        RTS
FNDADR  STA   ADRCHR     ;Calcule adr. en mem
        LDA   £0         ;matrice du caractere
        STA   ADRCHR+1   ;dont code dans A
        CLC              ;resultat dans INDCHR
        ROL   ADRCHR     ;Formule:
        ROL   ADRCHR+1   ;ADRCTBL + (A*8)
        ROL   ADRCHR
        ROL   ADRCHR+1
        ROL   ADRCHR
        ROL   ADRCHR+1
        LDA   ADRCTBL
        CLC
        ADC   ADRCHR
        STA   ADRCHR     ;Resultat dans INDCHR
        STA   ADRCHR2    ;et dans les pointeurs
        STA   INDCHR     ;ADRCHR,ADRCHR2
        TAX
        LDA   ADRCTBL+1
        ADC   ADRCHR+1
        STA   ADRCHR+1
        STA   ADRCHR2+1
        STA   INDCHR+1
        TAY
        RTS
DSHIFT  LDX   £0
        STX   LINNUM
        STX   LINNUM+1
        LDX   £7
SHFT1   ASL
        PHP
        JSR   ROLINNUM
        PLP
        JSR   ROLINNUM
        DEX
        BPL   SHFT1
        JSR   ROLINNUM   ;Libere bit 7 1er oct.
        LSR   LINNUM
        RTS
ROLINNUM ROL  LINNUM     ;Decale 16 bits LINNUM
        ROL   LINNUM+1   ;Carry -> bit 0
        RTS
*******************************
ENDFILE EQU   *+$800
```

# Fichier 'ASCII.SET'

*Après avoir saisi ce fichier sous moniteur, vous le sauvegarderez par :*

BSAVE ASCII.SET,A$5100,L$200

```
5100:00 00 00 00 00 00 00 00
5108:08 08 08 08 08 00 08 00
5110:14 14 14 00 00 00 00 00
5118:14 14 3E 14 3E 14 14 00
5120:08 3C 0A 1C 28 1E 08 00
5128:06 26 10 08 04 32 30 00
5130:04 0A 0A 04 2A 12 2C 00
5138:08 08 08 00 00 00 00 00
5140:08 04 02 02 02 04 08 00
5148:08 10 20 20 20 10 08 00
5150:08 2A 1C 08 1C 2A 08 00
5158:00 08 08 3E 08 08 00 00
5160:00 00 00 00 08 08 04 00
5168:00 00 00 3E 00 00 00 00
5170:00 00 00 00 00 00 08 00
5178:00 20 10 08 04 02 00 00
5180:1C 22 32 2A 26 22 1C 00
5188:08 0C 08 08 08 08 1C 00
5190:1C 22 20 18 04 02 3E 00
5198:3E 20 10 18 20 22 1C 00
51A0:10 18 14 12 3E 10 10 00
51A8:3E 02 1E 20 20 22 1C 00
51B0:38 04 02 1E 22 22 1C 00
51B8:3E 20 10 08 04 04 04 00
51C0:1C 22 22 1C 22 22 1C 00
51C8:1C 22 22 3C 20 10 0E 00
51D0:00 00 08 00 08 00 00 00
51D8:00 00 08 00 08 08 04 00
51E0:10 08 04 02 04 08 10 00
51E8:00 00 3E 00 3E 00 00 00
51F0:04 08 10 20 10 08 04 00
51F8:1C 22 10 08 08 00 08 00
5200:10 20 1C 20 3C 22 3C 00
5208:08 14 22 22 3E 22 22 00
5210:1E 22 22 1E 22 22 1E 00
5218:1C 22 02 02 02 22 1C 00
5220:1E 22 22 22 22 22 1E 00
5228:3E 02 02 1E 02 02 3E 00
5230:3E 02 02 1E 02 02 02 00
5238:3C 02 02 02 32 22 3C 00
5240:22 22 22 3E 22 22 22 00
5248:1C 08 08 08 08 08 1C 00
5250:20 20 20 20 20 22 1C 00
5258:22 12 0A 06 0A 12 22 00
5260:02 02 02 02 02 02 3E 00
5268:22 36 2A 2A 22 22 22 00
5270:22 22 26 2A 32 22 22 00
5278:1C 22 22 22 22 22 1C 00
5280:1E 22 22 1E 02 02 02 00
5288:1C 22 22 22 2A 12 2C 00
5290:1E 22 22 1E 0A 12 22 00
5298:1C 22 02 1C 20 22 1C 00
52A0:3E 08 08 08 08 08 08 00
52A8:22 22 22 22 22 22 1C 00
52B0:22 22 22 22 22 14 08 00
52B8:22 22 22 2A 2A 36 22 00
52C0:22 22 14 08 14 22 22 00
52C8:22 22 14 08 08 08 08 00
52D0:3E 20 10 08 04 02 3E 00
52D8:3E 06 06 06 06 06 3E 00
52E0:00 00 3C 02 02 3C 10 08
52E8:3E 30 30 30 30 30 3E 00
52F0:00 00 08 14 22 00 00 00
52F8:00 00 00 00 00 00 00 7F
```

## Programme 'DEMO.START'

```
10  PRINT  CHR$ (4)"PR£3"
20  PRINT  CHR$ (4)"BRUNDHGR
25  PRINT  CHR$ (4)"BLOADASCII.SET,A$5100"
27  POKE 768,0: POKE 769,80
30  &  HGR : TEXT
35  POKE  - 16368,0: PRINT "DHGR: Le Basic Do
    uble-Resolution"
40  PRINT : PRINT : PRINT "Appuyez sur une to
    uche pour continuer ---> ";: GET A$
50  HOME : PRINT
60  PRINT  CHR$ (4)"RUNDEMO"
```

## Programme 'DEMO'

```
5  TEXT
10  PRINT  CHR$ (4)"PR£3"
15  PRINT
30  &  HGR2 : &  LET 0
34  POKE 772,0: &  PRINT 5,30,"Pensez-vous av
    oir atteint les limites de votre Apple?"
36  FOR I = 1 TO 1500: NEXT : &  PRINT 15,60,
    "Eh bien non! Car voici..."
40  POKE 772,128: &  DEF 96;62,127,127,127,12
    7,127,62,8
50  FOR I = 1 TO 2500: NEXT
60  FOR I = 191 TO 140 STEP  - 1: &  PRINT 30
    ,I,"` D H G R `": NEXT
65  POKE 772,0: FOR I = 191 TO 160 STEP  - 1:
     &  PRINT 34,I,"DOUBLE": NEXT
67  FOR I = 191 TO 174 STEP  - 1: &  PRINT 37
    ,I,"GRAPHICS": NEXT
90  FOR I = 1 TO 3000: NEXT
100  &  CLEAR  INVERSE : &  CLEAR 0,255: &  C
    LEAR 0,0
105  &  HGR2
110  FOR I = 15 TO 0 STEP  - 1
120  &  HOME 20,59,0,1 + I
130  &  CLEAR 16 * I + I,16 * I + I: NEXT
140  &  HOME 0,79,0,23
150  POKE 772,128
160  &  PRINT 30,160,"D  H  G  R"
170  POKE 772,0: &  PRINT 35,175,"Par Pascal
    CANTOT"
180  GOSUB 900: &  HPLOT 0,148 TO 559,148
183  &  HCOLOR= 0: &  LET 2: &  PRINT 30,132,
    "DOUBLE GRAPHICS": &  LET 0
185  FOR I = 1 TO 280: &  HPLOT 140 + I,1 TO
    140,I / 2 + .5: NEXT
186  &  PRINT 22,30,"DE NOUVELLES": &  PRINT
    25,50,"POSSIBILITES"
187  FOR I = 140 TO 1 STEP  - 1: &  HPLOT 420
    ,I TO 420 - I * 2,I: NEXT
188  &  PRINT 37,85,"DE NOUVEAUX": &  PRINT 4
    0,105,"HORIZONS"
195  FOR I = 1 TO 5000: NEXT
200  &  GR 5
220 X = 20:C = 68: GOSUB 920
222 X = 30:C = 72: GOSUB 920
224 X = 40:C = 71: GOSUB 920
226  FOR X = 79 TO 50 STEP  - 1: &  COLOR= 7:
     &  DRAW 82 AT X,20
228  POKE  - 16336,0: IF X > 50 THEN  &  COLO
    R= 5: &  DRAW 82 AT X,20
229  NEXT : PRINT  CHR$ (7);: &  COLOR= 15: &
     DRAW 82 AT 50,20
240  VTAB 22: PRINT "VOICI ENFIN UN BASIC DOU
    BLE-HAUTE RESOLUTION !!!"
245  PRINT : PRINT "Pour Apple //e 128K, Appl
    e //c et Apple //gs";
290  FOR I = 1 TO 5000: NEXT
300  FOR C = 15 TO 0 STEP  - 1: &  GR C: FOR
    I = 1 TO 500: NEXT I,C
310  &  HGR : POKE 772,128: &  PRINT 36,74,"P
    OM'S"
315  POKE  - 16368,0: PRINT : PRINT : PRINT :
     PRINT "Appuyez sur une touche pour conti
    nuer... "
318  FOR C = 15 TO 0 STEP  - 1: &  HCOLOR= C
320  FOR I = 0 TO 559 STEP 20: &  HPLOT 559 -
     I,0 TO 0,I / 3.5: &  HPLOT 559 - I,160 T
    O 0,160 - I / 3.5: &  HPLOT I,160 TO 559,
    160 - I / 3.5: &  HPLOT I,0 TO 559,I / 3.
    5: NEXT
325  IF ( PEEK ( - 16384) > 128) AND C THEN C
     = 1: POKE  - 16368,0
330  NEXT
390  FOR I = 1 TO 2000: NEXT
400  &  HGR2 : GOSUB 900: POKE 772,0
410  &  PRINT 20,100,"Et même un Basic graphi
    que bien amélioré:"
412  &  PRINT 30,120,"- Générateur de caractè
    res 2 tailles en DHGR"
414  &  PRINT 30,128,"- Caractères graphiques
     couleurs en Basse-résol."
416  &  PRINT 30,136,"- Fonction SCRN en DHGR
    "
418  &  PRINT 30,144,"- Instruction de fenètr
    age"
420  &  PRINT 30,152,"- Couleurs mixtes pour
    le fond"
422  &  PRINT 30,160,"- Pseudo-couleur invers
    e"
424  &  PRINT 30,168,"- etc!
430  &  HOME 30,50,3,10
440  FOR I = 0 TO 30:C1 =  INT ( RND (1) * 16
    ):C2 =  INT ( RND (1) * 16)
445  &  CLEAR C1 + C1 * 16,C2 + C2 * 16
447  FOR J = 1 TO 500: NEXT
450  NEXT
460  &  CLEAR 255,255: POKE 772,128: INVERSE
    : &  PRINT 36,50,"DHGR": NORMAL
465  &  HCOLOR=  INVERSE
470  FOR I = 0 TO 191 STEP 4: &  HPLOT 0,I TO
     559,I
471  FOR J = 1 TO 100: NEXT
472  &  HPLOT 0,I TO 559,I: NEXT
473  HOME : FOR I = 1 TO 5000: NEXT
890  TEXT : PRINT  CHR$ (4)"PR£3": HOME : END
900  &  HCOLOR= 15: &  HPLOT 0,0 TO 559,0 TO
    559,191 TO 0,191 TO 0,0
910  RETURN
920  FOR I = 1 TO 20: &  COLOR= 7: &  DRAW C
    AT X,I
922  POKE  - 16336,0: IF I < 20 THEN  &  COLO
    R= 5: &  DRAW C AT X,I
925  NEXT : PRINT  CHR$ (7);: &  COLOR= 15: &
     DRAW C AT X,I - 1
930  RETURN
```

## Récapitulation 'DHGR'

*Après avoir saisi ce code sous moniteur, vous le sauvegarderez par :*

```
BSAVE DHGR,A$4000,
      L$7F5
```

```
4000:4C 45 40 4C D4 41 4C 63
4008:42 4C 24 42 4C A8 42 4C
4010:B7 42 4C 64 FB 4C F0 42
4018:4C A9 43 4C BB 43 4C 54
4020:45 4C AA 44 4C 53 44 4C
4028:71 43 4C DD 43 4C EA 46
4030:4C 89 46 4C 98 47 4C 2F
4038:46 4C A1 41 4C A1 41 4C
4040:A1 41 4C A1 41 A0 02 B9
4048:51 40 99 F5 03 88 10 F7
4050:60 4C 59 40 A5 2E 4C A1
4058:41 20 87 00 85 2E A2 00
4060:BD 81 40 F0 EF C5 2E F0
4068:03 E8 10 F4 8A 0A AA 8D
4070:93 40 8D 7F 40 BD 94 40
4078:8D 80 40 20 81 00 4C C7
4080:42 88 8D A0 8E 8F 93 BA
4088:92 8E B8 91 90 94 97 BD
4090:AA D5 00 13 41 4F 41 8C
4098:41 55 41 70 41 6C 44 38
40A0:46 59 43 22 41 87 43 A9
40A8:43 BB 43 E3 42 CA 40 C1
40B0:43 25 46 B5 40 20 F8 E6
40B8:E0 04 B0 54 86 18 8A 0A
40C0:65 18 69 39 8D C8 40 4C
40C8:39 40 20 F8 E6 E0 50 B0
40D0:3F 86 19 20 4C E7 E0 50
40D8:B0 36 E4 19 90 32 8A 38
40E0:E5 19 85 18 20 4C E7 8E
40E8:A6 41 E0 18 B0 22 20 4C
40F0:E7 E0 18 B0 1B EC A6 41
40F8:90 16 86 23 AD A6 41 85
4100:22 A5 18 85 21 A5 19 85
4108:20 E6 21 E6 23 4C C1 FB
4110:4C 99 E1 A5 30 8D A7 41
4118:20 8C 41 A9 14 85 22 4C
4120:DC 41 20 87 00 48 20 B1
4128:00 68 C9 88 F0 07 C9 91
4130:F0 14 4C A1 41 20 AB 41
4138:20 24 42 48 20 BE DE 68
4140:A8 A9 00 4C CA 41 20 8D
4148:45 20 53 44 38 80 EC 20
4150:AB 41 4C 63 42 20 BA 41
4158:86 1A 20 BE DE 20 BA 41
4160:86 19 20 97 41 20 C2 41
4168:8A A6 19 A4 1A 4C A8 42
4170:20 C2 41 86 1A 20 BE DE
4178:20 C2 41 86 19 20 97 41
4180:20 BA 41 8A A8 A5 1A A6
4188:19 4C B7 42 20 F8 E6 E0
4190:10 B0 15 8A 4C 64 F8 20
4198:87 00 C9 C5 D0 03 4C B1
41A0:00 A2 10 4C 12 D4 00 00
41A8:4C 99 E1 20 BA 41 86 19
4180:20 BE DE 20 C2 41 8A A4
41B8:19 60 20 F8 E6 E0 50 B0
41C0:E7 60 20 F8 E6 E0 30 B0
41C8:DF 60 20 F2 E2 20 E3 DF
41D0:AA 4C 2B EB A5 30 8D A7
41D8:41 20 64 F8 8D 50 C0 8D
41E0:0D C0 8D 7E C0 8D 5E C0
41E8:8D 53 C0 8D 56 C0 8D 00
41F0:CD A5 30 85 19 20 C8 42
41F8:A9 00 85 18 A5 18 20 47
4200:FB A5 30 A0 27 8D 05 C0
4208:A5 30 91 26 8D 04 C0 A5
4210:19 91 26 88 10 EF E6 18
4218:A5 18 C9 14 90 DE AD A7
4220:41 85 30 60 48 A9 91 8D
4228:9C 42 68 20 63 42 A9 B1
4230:8D 9C 42 A5 2E 49 FF 85
4238:19 AD A7 41 25 2E C9 10
4240:90 04 4A 4A 4A 4A 2C 07
4248:03 10 13 A0 00 85 19 0A
4250:0A 0A 0A 05 19 D9 D3 42
4258:F0 03 C8 10 F8 98 29 0F
4260:C9 01 60 4A 08 4E 07 03
4268:20 47 F8 28 A5 30 48 A9
4270:0F 90 02 A9 F0 85 2E 98
4278:48 8D 01 C0 8D 54 C0 4A
42B0:A8 B0 0E 38 6E 07 03 8D
4288:55 C0 98 48 20 C8 42 68
4290:A8 B1 26 8D A7 41 45 30
4298:25 2E 51 26 91 26 8D 54
42A0:C0 68 A8 68 85 30 18 60
42A8:86 2C 20 63 42 C4 2C 80
4280:16 C8 20 77 42 90 F6 86
4288:2D 38 B0 02 69 01 48 20
42C0:63 42 68 C5 2D 90 F5 60
42C8:A5 30 29 0F A8 89 D3 42
42D0:85 30 60 00 88 11 99 22
42D8:AA 33 BB 44 CC 55 DD 66
42E0:EE 77 FF 20 F8 E6 86 18
42E8:20 97 41 20 A8 41 A6 18
42F0:8E 8A 47 8D 8C 47 8C 88
42F8:47 84 50 8A 20 98 47 20
4300:8F 47 A9 08 85 18 AD 00
4308:20 4D 8D 47 A2 08 8E A6
4310:41 A4 50 8C 8B 47 4A 90
4318:13 48 AD 8C 47 AC 8B 47
4320:C9 30 BD 07 C0 50 80 03
4328:20 63 42 68 EE 88 47 CE
4330:A6 41 D0 E2 EE 8C 47 EE
4338:07 43 D0 03 EE 08 43 C6
4340:18 D0 C3 60 38 24 18 8D
4348:5E C0 8D 50 C0 8D 57 C0
4350:8D 53 C0 90 03 8D 52 C0
4358:60 20 B7 00 C9 9E D0 06
4360:6E 05 03 4C B1 00 C9 9D
4368:D0 03 18 90 F3 20 FB E6
4370:8A C9 FF F0 EB 4E 05 03
4378:8D 86 43 0A 0A 0A 0A 0D
4380:86 43 8D 86 43 60 FF 20
4388:F8 E6 8A 20 98 47 A9 3B
4390:20 C0 DE A9 07 85 18 D0
4398:03 20 BE DE 20 F8 E6 A4
43A0:18 8A 91 CE C6 18 10 F1
43A8:60 8D 00 C0 8D 05 C0 20
43B0:E2 F3 8D 04 C0 20 E2 F3
43B8:4C 46 43 20 A9 43 4C 44
43C0:43 4E 06 03 20 B7 00 C9
43C8:9E D0 09 6E 06 03 20 81
43D0:00 4C DD 43 20 F8 E6 8A
43D8:48 20 4C E7 68 86 51 85
43E0:50 A5 23 0A 0A 0A 8D A6
43E8:41 A5 20 18 65 21 8D 51
43F0:44 A5 22 0A 0A 0A 8D 52
43F8:44 8D 5E C0 AD 52 44 20
4400:A7 45 A5 20 85 18 A0 00
4408:AD 52 44 4A 90 01 C8 B9
4410:50 00 85 19 8D 01 C0 8D
4418:54 C0 A5 18 4A B0 03 8D
4420:55 C0 AB 2C 06 03 30 1F
4428:A5 19 91 26 2A 26 19 E6
4430:18 A5 18 CD 51 44 90 DF
4438:EE 52 44 AD 52 44 CD A6
4440:41 90 B9 8D 54 C0 60 B1
4448:26 49 7F 91 26 38 B0 DF
4450:00 00 00 48 A9 81 8D 82
4458:45 68 20 54 45 A2 91 8E
4460:82 45 25 30 F0 04 A9 01
4468:38 24 18 60 20 B7 00 C9
4470:C1 F0 0F 20 8D 45 20 54
4478:45 20 87 00 F0 2B C9 C1
4480:D0 27 20 B1 00 20 8D 45
4488:8E A6 44 8C A7 44 8D A8
4490:44 84 9D A8 8A A6 9D 20
4498:AA 44 AE A6 44 AC A7 44
44A0:A0 A8 44 4C 76 44 00 00
44A8:00 60 48 38 E5 E0 48 8A
44B0:E5 E1 85 D3 B0 0A 68 49
44B8:FF 69 01 48 A9 00 E5 D3
44C0:85 D1 85 D5 68 85 D0 85
44C8:D4 68 85 E0 86 E1 98 18
44D0:E5 E2 90 04 49 FF 69 FE
44D8:85 D2 84 E2 66D3 38 E5
44E0:D0 AA A9 FF E5 D1 85 1D
44E8:A4 E5 B0 05 0A 20 22 45
44F0:38 A5 D4 65 D2 85 D4 A5
44F8:D5 E9 00 85 D5 08 B4 E5
4500:20 5A 45 A4 E5 28 E8 D0
4508:04 E6 1D F0 14 A5 D3 B0
4510:D8 20 D3 F4 18 A5 D4 65
4518:D0 85 D4 A5 D5 65 D1 50
4520:DA 60 10 13 A5 30 4A B0
4528:05 49 C0 85 30 60 88 10
4530:02 A0 4F A9 C0 D0 13 A5
4538:30 0A 49 80 10 03 85 30
4540:60 A9 81 C8 C0 50 90 02
4548:A0 00 84 E5 85 30 08 20
4550:05 46 28 60 8D 00 C0 20
4558:A7 45 46 E5 A5 E5 8D 01
4560:C0 8D 54 C0 B0 03 8D 55
4568:C0 A4 E5 2C 05 03 10 07
4570:81 26 45 30 4C 80 45 AD
4578:50 44 51 26 25 30 51 26
4580:29 7F 91 26 8D 54 C0 8D
4588:00 C0 26 E5 60 20 67 DD
4590:20 52 E7 A4 51 A6 50 C0
4598:02 90 06 D0 07 E0 30 80
45A0:03 4C CD F6 4C 99 E1 86
45A8:E0 84 E1 C5 E2 F0 23 85
45B0:E2 48 29 C0 85 26 4A 4A
45B8:05 26 85 26 68 85 27 0A
45C0:0A 0A 26 27 0A 26 27 0A
45C8:66 26 A5 27 29 1F 05 E6
45D0:85 27 8A C0 00 F0 22 88
45D8:F0 06 A0 48 69 01 D0 18
45E0:C9 FB 90 10 C9 FF D0 06
45E8:A9 F9 A0 49 D0 0F 69 01
45F0:A0 48 D0 09 A0 23 69 05
45F8:C8 E9 07 B0 FB 84 E5 AA
4600:BD B9 F4 85 30 98 29 0F
4608:86 19 AA AD 86 43 F0 0F
4610:C9 FF F0 08 E0 00 F0 07
4618:48 2A 68 2A CA D0 F9 8D
4620:50 44 A6 19 60 20 F8 E6
4628:E0 05 90 03 4C 99 E1 BD
4630:36 46 8D 79 47 60 91 11
4638:51 31 81 20 F8 E6 E0 50
4640:B0 2A 8E 6A 46 20 8E DE
4648:20 F8 E6 E0 C0 B0 1D 8E
4650:6B 46 20 BE DE 20 87 00
4658:F0 0F 90 13 20 7D E0 B0
4660:0E C9 22 F0 0A C9 BB F0
4668:EC 60 00 00 4C 99 E1 A9
4670:00 85 52 20 7B DD 24 11
4678:D0 09 20 34 ED 20 92 46
4680:4C 55 46 20 95 46 4C 55
4688:46 8E 6A 46 8C 68 46 4C
4690:9B 46 20 E7 E3 20 00 E6
4698:AA F0 CE A0 00 81 5E F0
46A0:C8 20 AB 46 CA F0 C2 C8
46A8:D0 F3 60 8D 8A 47 98 48
4680:8A 48 AE 6A 46 AD 68 46
46B8:A8 AD 8A 47 20 EA 46 EE
46C0:6A 46 2C 04 03 10 03 EE
46CB:6A 46 AD 6A 46 C9 50 90
46D0:14 A9 00 8D 6A 46 AD 68
46D8:46 18 69 08 C9 C0 90 02
46E0:A9 00 8D 68 46 68 AA 68
46E8:A8 60 8D 8A 47 8E 8B 47
46F0:8C 8C 47 8E A6 41 AD 8C
46F8:47 4E 8E 47 A0 00 AD 8A
4700:47 20 8F 47 4E A6 41 6E
4708:8E 47 AD 8A 47 20 98 47
4710:A2 08 86 18 AD 8C 47 C9
4718:C0 80 16 20 A7 45 AC A6
4720:41 AD 00 20 4D 8D 47 29
4728:7F 2C 04 03 30 17 20 69
4730:47 EE 8C 47 20 81 47 C6
4738:18 D0 D9 AD 8A 47 AE 8B
4740:47 AC 8C 47 60 20 D6 47
4748:A5 50 20 69 47 AD 8E 47
4750:48 49 80 8D 8E 47 30 05
4758:C8 C0 28 B0 05 A5 51 20
4760:69 47 68 8D 8E 47 4C 31
4768:47 8D 01 C0 8D 54 C0 2C
4770:8E 47 30 03 8D 55 C0 09
4778:80 91 26 91 26 8D 54 C0
4780:60 EE 22 47 D0 03 EE 23
4788:47 60 00 00 00 00 00 A2
4790:00 24 32 30 02 A2 FF 8E
4798:8D 47 60 8D 22 47 A9 00
47A0:8D 23 47 18 2E 22 47 2E
47A8:23 47 2E 22 47 2E 23 47
4780:2E 22 47 2E 23 47 AD 00
4788:03 18 6D 22 47 8D 22 47
47C0:8D 07 43 85 CE AA AD 01
47C8:03 6D 23 47 8D 23 47 8D
47D0:08 43 85 CF AB 60 A2 00
47D8:86 50 86 51 A2 07 0A 08
47E0:20 F0 47 28 20 F0 47 CA
47E8:10 F4 20 F0 47 46 50 60
47F0:26 50 26 51 60
```


Reliures Pom's

60,00 F franco, bon de commande page 71

Pour six numéros

# les infos Apple // for ever

## Jean-Yves Bourdin

Note : Nous utiliserons dans cette chronique et dans les prochaines l'expression "ProDOS (8 ou 16) standard" pour désigner ces logiciels ProDOS non protégés, qu'on peut mettre dans n'importe quel sous-catalogue, configurables, qui acceptent les CDA et renvoient au Quit du ProDOS en sortie.

## Nouvelles du front

Certains processeurs-carbone (autrefois appelés cerveaux) doivent avoir une vitesse d'horloge un peu trop rapide. Ce n'est pas parce que j'ai écrit dans Pom's n° 34 que *«la technologie 8 bits seule est une technologie dépassée»*, ce qui est un fait aussi bien pour l'IBM que pour l'Apple //, qu'il faut jeter votre //c à la poubelle. La technologie n'est pas tout, et l'Apple // sera dépassé quand il ne sortira plus de logiciels pour lui. Lisez *Apple // for ever*, et vous verrez que ce n'est pas demain la veille. Ce n'est pas non plus parce que le 65832 est en cours de développement, tout comme les accélérateurs GS, que nous allons les avoir demain. Apple a précisé tout à fait officiellement qu'il ne sortirait aucune nouvelle machine en 88. Pas de panique donc.

Apple // for ever, cela ne signifie ni 'Apple // Plus for ever' ni 'Le 32 bits 20 Mhz tout de suite'. Cela signifie seulement que l'Apple // ne mourra pas comme est mort par exemple l'Atari 8 bits, veuf et sans héritiers (vous qui êtes tentés par une autre machine, ayez une pensée pour cet Atari-là). Le relais de l'Apple // est pris, par le GS, et le relais du GS sera pris un jour, par une nouvelle machine 32 bits, intégrant encore mieux Mac et Apple //, accélérée, etc. C'est cette durabilité qui a été invoquée comme raison décisive par l'armée américaine pour acheter 4 000 IIGS d'un coup pour ses écoles, quitte à diffuser chez les 'GI' un *esprit Wozniak* qui n'est pas précisément militariste.

Apple // for ever, c'est peut-être cette phrase de Bill Atkinson à propos de son logiciel Hypercard qui agite tant, et à juste titre, le monde du Mac en ce moment : *«J'aimerais également voir une version de ce produit sur l'Apple //, ce qui ne pose aucun problème au sein de la société Apple, et je suis prêt à aider le programmeur qui souhaite le faire»* (Interview dans l'ouvrage de Danny Goodman 'The Complete Hypercard Handbook', chez Bantam).

Apple // for ever, c'est enfin la bonne santé de la société Apple. Quelques chiffres : au cours du dernier trimestre 1987, Apple USA a réalisé un chiffre d'affaires d'un milliard de dollars. Les analystes financiers prévoient un chiffre d'affaires de trois milliards et demi de dollars pour l'année 1988. Au cours du seul mois de décembre 1987, Apple USA a vendu 40 000 IIGS. Précisons encore que si pour l'ensemble de l'année 1987, les utilisateurs américains ont acheté 145 467 IIGS, 109 680 d'entre-eux ont encore voulu à tout prix acheter un Apple //c et 75 298 un Apple //e...

La deuxième Fête de l'Apple // (ni Mac World ni Apple Expo, une fête organisée par et pour les utilisateurs de l'Apple //), après le succès de celle de 87 à San Francisco, se tiendra à Boston du 20 au 22 mai. Si vous passez aux États-Unis à cette saison, saluez Wozniak de ma part (il y sera), et tâchez de savoir où en est Hypercard pour Apple //. Mais pensez à vous inscrire à l'avance, on attend la foule.

Une autre bonne nouvelle : **Activision**, qui diffuse entre autres les versions US de GSWrite et GSPaint, a décidé de mettre au standard ProDOS ses logiciels de "productivité" et de "créativité" pour GS (mais pas les jeux). Il faut dire que la protection rendait les programmes totalement inutilisables sur les GS dotés d'un lecteur 3'5 Unidisk Apple, ou de la carte UDC de Central Point. Et que **Copy ][ Plus** de **Central Point Software**, ainsi que **Disk Util //** de **FWB** (tous deux ProDOS standard), contiennent les paramètres nécessaires pour déprotéger les programmes GS sur 3'5. **Data Pak Software** a également mis au standard ProDOS la version 2.0 de **Graphic Writer** pour GS. Il est bon que la spirale protection/copie ne recommence pas sur les 3'5. Et cela n'empêche pas Activision de réaliser des profits confortables : 524 000 dollars de bénéfices déclarés sur un chiffre d'affaires de 8 700 000 dollars pour la dernière année fiscale.

D'autres, comme **Styleware** (Multiscribe GS 3.0), ou **Electronic Arts** (Deluxe Paint //), ont une autre formule : la disquette que vous achetez chez votre vendeur est protégée, mais en renvoyant 10 ou 20 dollars avec votre carte d'enregistrement à l'éditeur, vous avez une version ProDOS standard (mais numérotée, avec vos coordonnées sur la disquette). Mon conseil : incluez ce supplément dans votre budget pour ce logiciel, et considérez la version protégée comme une version de test, juste pour voir si le programme marche sur votre configuration. Attendez pour utiliser régulièrement le programme de pouvoir travailler

avec une copie de la version standard. Cette formule est un peu compliquée, mais elle a sans doute de l'avenir, car elle permet à l'éditeur de lutter contre le piratage sans gêner l'utilisateur.

AppleWorks n'est plus un produit Apple : Apple ne vend plus de logiciels. Ceux qu'il réalise seront désormais donnés avec les machines. Les autres ne seront plus vendus ni supportés par Apple, mais par sa filiale **Claris**. Il s'agit de mettre les logiciels d'origine Apple sur un pied d'égalité avec les autres produits. Utilisateurs d'AppleWorks 1.4, c'est désormais à la dynamique société **P. Ingénierie**, qui est aussi **Claris France**, qu'il faut vous adresser.

Après toutes ces bonnes nouvelles, on me pardonnera un petit coup de griffe. Le fétichisme, dit le dictionnaire, est *«une forme de superstition primitive consistant à attribuer des pouvoirs humains ou surnaturels à des objets inanimés ou à de simples symboles»*. L'informatique, pourtant fille de la pensée rationnelle moderne, ne semble pas délivrée du fétichisme des logos, sigles et autres gris-gris, si l'on en juge par la bataille juridique engagée par **MicroPro**, éditeur du traitement de textes **WordStar**, contre **MicroProse**, éditeur de jeux comme **F15 Strike Eagle**. En conclusion de cette bataille, MicroProse s'engage à changer de sigle dans les deux ans, et MicroPro s'engage à ne jamais utiliser le sigle MicroProse. Devinez aussi pourquoi GS Com est devenu VS Com...

Dernière minute : une dépêche datée du 31 mars à minuit nous apprend que **Citroën** a déposé un recours contre **Apple France** pour lui faire renoncer au sigle «GS» que Citroën avait déposé pour ses automobiles... Le bruit court également que **Georges Marchais** et **Lionel Jospin**, réconciliés pour le second tour de l'élection présidentielle, auraient décidé d'attaquer ensemble **IBM France** pour utilisation abusive des sigles «PC» et «PS»...

## Toujours plus

Avec le DOS 3.3, ProDOS, CP/M, le Pascal UCSD, et j'en passe, les //e et //c n'avaient toujours pas accès à un système d'exploitation qui soit directement graphique-souris, sans l'intermédiaire d'un programme. C'est maintenant chose faite avec **GEOS**, de **Berkeley Softworks**. Pour 70 dollars, vous avez un système d'exploitation style Macintosh, un traitement de textes *Wysiwyg*, un logiciel graphique, des accessoires de bureau, un correcteur, des utilitaires de conversions de fichiers et d'impression (sur Laser entre autres). D'où sortent-ils, ceux-là ? Eh bien, il existe un ordinateur 6502 8 bits, technologiquement dépassé, mais auquel ses utilisateurs sont si attachés que les programmes continuent à sortir en grand nombre et à bon marché. Avec son moniteur intégré et l'Assembleur Merlin de Glen Bredon, on peut presque tout lui faire faire. Vous l'avez reconnu, c'est... le Commodore 64 ! La quantité de logiciels sous GEOS y est impressionnante : bienvenue sur Apple // !

Aveugles, lisez attentivement ceci. (*N.D.L.R. : excusez-le...*) Les cartes **Valentine** et **Sonolect** permettent de faire parler l'Apple // et de lui faire lire l'écran à haute voix. Avec le traitement de textes **Gribouille Sonolect**, vous avez un ensemble micro-informatique parfaitement fonctionnel, la carte Valentine semblant avoir atteint le niveau qu'on pouvait en attendre (lire notre essai dans le numéro 31 de Pom's). Prenez contact avec l'**Association Valentin Haüy** et le **Club Micro Son**.

À la sortie du câble couleur de votre //c, de la carte couleur de votre //e ou ][+, du câble GS/Péritel, il y a une prise Péritel qui vous permet de le brancher sur votre téléviseur, d'accord ? Maintenant, regardez bien votre magnétoscope : tiens, il a aussi une entrée Péritel ! Vos souvenirs de vacances sur vidéo-cassettes sont sympas, mais manquent de présentation ? Votre vidéothèque comporte d'indésirables séquences sur les lessives au milieu des films ? Enregistrez donc les images de votre Apple ! Avez-vous essayé le kaléidoscope de **Print Shop (Broderbund)** ou les images de **Deluxe Paint (Electronic Arts)** sur vos cassettes ? Il y a bien entendu au moins un éditeur qui s'est rendu compte que l'Apple // (pas le Mac Plus, hein coco, l'Apple // seulement) était un excellent générateur de banc-titres, c'est **Electronic Arts** avec **Video Title Shop**. Mais tous les programmes graphiques et toutes les images en couleurs conviennent. Et **Beagle Screens** de **Beagle Bros** permet des présentations très drôles, avec même de charmantes petites animations.

Vous pouvez d'ailleurs faire bien d'autres choses avec un magnétoscope, une caméra, un téléviseur, un lecteur de disques compacts et un Apple // : nous en parlerons une prochaine fois. En attendant, regardez donc si votre magnétoscope n'aurait pas une télécommande mécanique, qu'on pourrait remplacer, par exemple, par un cordon qui le relierait à l'Apple //...

La Publication Assistée par Ordinateur (PAO) n'est pas une exclusivité du Mac (même s'il faut reconnaître qu'il y est le meilleur). Pom's consacrera un jour une étude à la PAO sur Apple //. Je me contenterai donc de mentionner ici trois logiciels qui font désormais partie des indispensables :

- **Print Magic**, d'**Epyx**, est le premier vrai concurrent (successeur ?) de **Print Shop** de **Broderbund**. Interface graphique-souris, bureau et accessoires, même sur un ][+, il utilise les polices de caractères de **Fontrix**, les dessins Apple, **Print Shop** ou **Newsroom**, est doté d'un excellent atelier graphique, permet de faire énormément de choses avec facilité et élégance. Bien entendu, il n'est pas protégé. Dans ces conditions, je n'ai trouvé qu'une critique à lui faire : pourquoi diable fonctionne-t-il sous DOS 3.3, et non sous ProDOS ?
- The **Graphics Manager**, de **On Three**, (ProDOS 8 standard)

vous permet d'organiser votre page imprimée point par point en graphique. Vous pouvez imprimer tous les dessins (HGR et DHGR) que vous voulez, dans la taille que vous voulez (agrandissement et réduction à volonté), à la place que vous voulez sur la page. Je ne me sers plus que de lui pour imprimer mes graphiques.

• **Publish-It !**, de **Timeworks** (ProDOS 8 standard), est vraiment PageMaker sur Apple //. Il utilise l'interface Mac, les polices de caractères Mac/GS (y compris sur //c et //e), et même la programmation *orientée-objet* du Mac. Vous pouvez utiliser vos six disquettes 3'5 de polices Mac/GS du domaine public diffusées par l'Apple Coop avec Publish-It à condition de changer leur type de $C8 en $F7 (pourquoi cette bizarrerie ?). Il permet une mise en pages impeccable des dessins et du texte, et offre une impression d'excellente qualité. En attendant un **Springboard Publisher** qui, à l'heure où j'écris ces lignes, fait toujours partie des programmes de l'Arlésienne, ce logiciel est véritablement une raison suffisante pour avoir un Apple //. Une seule critique, la même que celle d'Éric Weyland dans Pom's 34 à propos de Time Out Superfonts : vivement l'éditeur de polices GS !

## Le mange-disques

Les logiciels Domaine Public pour Apple // ont souvent une mauvaise réputation, à cause de l'héritage, qui fait encore circuler énormément de disquettes préhistoriques. Mais si on cherche dans les disquettes récentes, on peut trouver de l'or, aussi bien dans le domaine public proprement dit que dans le 'shareware' (mode de diffusion de logiciels par copie libre d'une version parfaitement fonctionnelle du programme : on essaye le programme et on paye ensuite directement à l'auteur). En voici une liste partielle, essentiellement puisée dans les logiciels de **l'Apple Coop** et du **Big Red Computer Club** (BRAC). Si certains vous intéressent, dites-le nous : si vous êtes suffisamment nombreux à le souhaiter, Pom's pourra peut-être en organiser la diffusion, comme nous le faisons déjà pour le Mac.

**Apple IIGS Fonts** : six disquettes 3'5 de polices Mac pour le GS, mais aussi pour //e et //c (avec des programmes comme Time Out Superfonts, Publish-It !, ou 816 Paint //e), **Apple Coop**. NB : tous les autres programmes énumérés ci-dessous sont en 5'25.

**Diversi-Copy 3.5, Diversi-Cache, Diversi-Key, Diversi-Hack : Share-ware** de **Bill Basham**, indispensables à tous les possesseurs de GS (voir Apple // for ever de Pom's 34), **BRAC**.

**Diversi-Dos version 4.1-C**, la dernière version du DOS ultra-complet et ultra-rapide de **Bill Basham**, shareware, **BRAC**.

**DiskWorks**, excellent éditeur de blocs sur tout disque ProDOS diffusé en shareware par **Living Legend Software**.

**ECP8**, interpréteur pour ProDOS style Unix, shareware, **BRAC**.

**MousePrintz**, qui patche Mousepaint pour le faire imprimer avec 80 imprimantes et 70 cartes d'interface différentes, et lui donner un bon paquet de possibilités supplémentaires, shareware, **BRAC**.

**Life Like**, un excellent "jeu de la vie" en ProDOS, shareware, **BRAC**.

**P8 Programs**, très bon disque domaine public du BRAC, contenant six utilitaires ProDOS indispensables.

**Squirt**, un sélecteur de programmes très honnête pour ProDOS 8, diffusé en shareware par le **BRAC** pour ceux qui ont le tort de ne pas avoir SuperMacroWorks et son **Bird's Better Bye**.

**BLU, Binary Linking Utility**, excellent utilitaire pour ajouter des routines "&" relogeables à tout programme Applesoft, diffusé en domaine public par le **BRAC** avec nombre de routines.

**FredWriter**, le traitement de textes complet sous ProDOS compatible AppleWriter, domaine public de **l'Apple Coop**.

**AMDOS 3.5**, un des meilleurs Dos 3.3 pour disques 3'5, de **Gary Little**, en shareware au **BRAC**.

**ImageWriter Utilities**, excellent disque sous DOS et ProDOS pour télécharger des polices, faire des copies d'écran, etc. sur Imagewriter I et II avec carte Super Série ou port //c (port GS exclu), domaine public du **BRAC**.

Sans compter une foultitude de disques de dessins et de bordures pour Print Shop, dont certains ont une qualité professionnelle, et tous ceux auxquels je ne songe pas en ce moment.

Pom's publiera peut-être un jour une étude sur les logiciels de décision graphique (camemberts, histogrammes, courbes, etc.) pour Apple //. Je me contenterai aujourd'hui de signaler (outre Time Out Graph, voir Pom's 34), trois très bons programmes récents au standard ProDOS :


# HyperPomme

Macintoshiste, Appledeuïste ! L'assistance téléphonique du Club Apple n'est plus ce qu'elle était mais les techniciens issus des SIG demeurent : ils se sont regroupés dans des associations régionales dont le but est de favoriser le contact entre les utilisateurs d'Apple et d'organiser des réunions.

Renseignements et inscriptions auprès de François Pirisi, **Fédération HyperPomme** au (1) 45 74 13 40 ou auprès de Jean Devaux, **HyperPomme Paris**, 16, bld Barbès 75018 Paris.

**Visualizer**, de **PBI**, existe en version GS et en version //e - //c. **Quick Chart**, de **Third Wave Technology**, et **Graphic Edge**, de **Pinpoint Publishing**, marchent en ProDOS 8 sur //e, //c, IIGS. J'avoue un faible pour Graphic Edge, pour sa programmation *orientée-objet*.

Ce bon vieil Applesoft : un des plus vieux Basics qui soient, des limites rédhibitoires... et pourtant non seulement vous et moi continuons à programmer avec lui, mais il dispose aujourd'hui d'une batterie de programmes qui en font, en fait, un excellent Basic. Deux explications à cela : d'abord, tout simplement, c'est le seul langage qui soit dans la Rom de tous les Apple //. Ensuite, c'est un langage ouvert, qu'on peut compléter, et qu'on connaît dans les moindres recoins (Merlin 8/16, de Glen Bredon chez Roger Wagner, est accompagné d'un programme qui sort sur imprimante 150 pages de listing source commenté de l'Applesoft qui est dans votre machine, y compris le relevé et l'explication de chaque bug).

Tâchons donc de voir avec quels outils nous allons continuer à programmer en Applesoft en 88. Comme il n'est pas question de faire un tour d'horizon complet de tous les utilitaires, je me contenterai de ceux qui sont à mon avis les meilleurs, les plus récents, et fonctionnent en ProDOS standard (la plupart fonctionnant aussi sous DOS).

Commençons par le commencement : l'éditeur. Vous êtes peut-être habitué à **GPLE** de **Beagle Bros** ou à **Gale** de **Microsparc** qui sont parmi les meilleurs. Ils sont aujourd'hui surpassés, par **Program Writer** de **Software Touch - Beagle Bros**, qui utilise des commandes et une interface type AppleWorks, permet les macro-commandes, et tout ce dont on peut avoir besoin pour écrire un programme. Surtout, sur //e, //c ou IIGS, il se loge dans un recoin (la carte langage de la mémoire auxiliaire, là où se loge aussi SuperMacroWorks) où il ne gêne nullement l'exécution du programme Basic : c'est strictement comme s'il ne prenait aucune place en mémoire. C'est d'ailleurs pourquoi il est le seul éditeur compatible avec le Beagle Compiler : pas étonnant, quand on sait que les deux ont le même auteur, Alan Bird.

Une fois votre programme écrit et testé, **D Code**, de **Beagle Bros**, vous aidera à tracer, comprimer, optimiser votre programme du point de vue de l'encombrement mémoire et de la vitesse. Je ne m'étendrai pas ici sur les fonctions de cet utilitaire indispensable, il me suffira de nommer son auteur : Alan Bird...

Enfin, votre programme une fois au point, vous pourrez avoir envie de le compiler pour accélérer sa vitesse d'exécution. Sous ProDOS, un seul programme vous le permet, de façon splendide : c'est le **Beagle Compiler** d'**Alan Bird** (encore lui), publié par **Beagle Bros** (encore eux). Je n'épiloguerai pas sur ce programme : si Pom's s'est 'décarcassé' pour diffuser ce programme en France, c'est qu'il y a une raison !

La seconde source, ce sont les extensions & qui sont écrites en code relogeable, qui peuvent fonctionner sans changement à n'importe quel emplacement mémoire (pour les techniciens : qui n'utilisent ni JMP ni JSR internes). Le plus souvent, ces routines sont collées de façon invisible au bout du programme Applesoft lui-même. J'en connais deux. D'abord le **BLU** (Binary Linking Utility) diffusé en domaine public par le **BRAC** (voir ci-dessus). Ensuite la série des Toolbox de Roger Wagner : **Wizard Toolbox**, **Database Toolbox**, **Video Toolbox**, **Chart and Graph Toolbox**. Avec cette série, vous pouvez pratiquement tout faire en Applesoft, sauf peut-être un serveur, mais pour cela il y a **Pom_Link**.

Je connais des utilisateurs qui cherchent désespérément le programme rarissime qui fera la chose qu'ils sont seuls au monde à vouloir faire sur l'imprimante qu'ils sont seuls au monde à utiliser avec la carte d'interface qu'ils ont. À tous ceux-là, la série des Toolbox offre la possibilité d'écrire eux-mêmes leur programme, ce qui est la seule solution.

Bien entendu, n'oublions pas qu'une des extensions les plus naturelles de l'Applesoft, ce sont les commandes supplémentaires ajoutées à Basic .System. Là encore, la première source, c'est la collection de Pom's : n'oubliez pas, vous avez les sources. J'y ajouterai une seconde source : **ProCommand** de **Glen Bredon**, diffusé par l'**Apple** Coop, non seulement parce qu'il offre une belle série de commandes, mais aussi un excellent éditeur très complet.

Il y a bien un logiciel qui pourrait dans une certaine mesure être comparé à un compilateur sous ProDOS. Il s'agit de Micol **Basic** de **Micol Systems**. Mais, et Micol est le premier à le dire, il ne s'agit pas d'un compilateur Applesoft, mais d'un autre Basic, incluant une grande partie de l'Applesoft, mais aussi de nombreuses fonctions nouvelles. Il faut programmer directement en Micol Basic, plutôt que réécrire un programme Applesoft existant, pour tirer profit de ce logiciel. Il serait terriblement lourd, long et encombrant pour qui voudrait l'utiliser comme compilateur Applesoft. En fait, c'est à d'autres "Super Basics", tels que **ZBasic** de **Zedcor**, qu'il doit être comparé (ce que nous ferons dans un prochain numéro).

Ceci dit, si vous souhaitez étendre les fonctions de l'Applesoft, une foule de programmes vous le permet : relisez la collection de Pom's, et vous constaterez que les commandes Applesoft les plus puissantes sont & et Call... Le problème en fait est de faire tenir ensemble tous ces ajouts hétéroclites. Je me limiterai donc ici aux programmes qui permettent la cohabitation de routines différentes.

La première source de ces extensions de l'Applesoft, c'est... la collection de Pom's. Non que les programmeurs

de Pom's soient tous des Alan Bird, mais réfléchissez simplement à ceci : avec Pom's, vous avez les sources des routines assembleur (même pas besoin de les taper, ils sont sur la disquette). Vous pouvez donc déplacer, modifier, ajuster, mettre ensemble ces routines. C'est quand même plus pratique que de passer des heures à désassembler avec Sourceror de Merlin, ce qui est la seule possibilité que nous laissent la plupart des programmes du commerce.

Terminons enfin par le meilleur, et de très loin, programme d'extension de l'Applesoft, un logiciel intégrateur qui vous permet d'utiliser n'importe quoi (lignes de Basic, routines assembleur ou autres) comme sous-routine ou comme procédure d'un programme Applesoft, avec gestion des erreurs, gestion de la pile, passage de paramètres, etc. Cet exploit de programmation s'appelle **Probasic**, et vous vous doutez de qui il est : d'Alan Bird bien sûr ! Le plus fort, c'est que ce programme n'est même pas vendu séparément par Beagle/Software Touch : il est mis en prime, comme ça, négligemment, sur la face 2 de **Program Writer** !

Un dernier mot pour une opinion personnelle : eh bien oui, je croyais, avant d'écrire ces lignes, que l'Applesoft était un Basic dépassé. Mais ce que ce travail m'a rappelé, c'est que cette notion purement technique de dépassement ne signifie rien du tout face à ce qu'Hervé Thiriez appelait *«l'inventivité des développeurs»* dans Pom's 34. Il se trouve que c'est l'Applesoft qui est dans la Rom de l'Apple //. Et il se trouve que c'est sur Apple // que programment les Glen Bredon et les Alan Bird. En conséquence, l'Applesoft est devenu un des Basic les plus puissants et les plus riches, à un point qui étonne. Gérer les outils du GS comme Quickdraw II avec un Applesoft qui ne connaît même pas la Double Haute Résolution, ce n'est pas un peu délirant, franchement ? Même Apple (sagement sans doute) ne s'y est pas risqué dans la Rom du GS. Eh bien ça existe, c'est dans Pom's 33 (**Q D Basic** de **LE Pham Hiep**) ! Alors... Applesoft for ever ?

# Sous le capot

Vous voulez connecter deux ordinateurs différents à trois imprimantes différentes, mettre une imprimante parallèle au bout d'un port série ou le contraire, relier votre ordinateur à plusieurs écrans, connecter une table traçante, une imprimante à aiguilles et une imprimante à marguerite sur le même port en choisissant par 'switch' entre les trois ? Bref, vous avez un problème de connectique ? C'est exactement le genre de choses qui fait perdre des heures et des jours de recherche, parce que c'est TOUJOURS techniquement possible, mais qu'on ne trouve JAMAIS le câble tout fait et même JAMAIS toutes les pièces nécessaires dans la même boutique ! Un jour vous remercierez Pom's de vous avoir trouvé l'adresse de la maison (aux USA) qui peut résoudre tous ces problèmes, et connecter n'importe quoi à n'importe quoi : **Support Systems International**. Si vous connaissez une adresse équivalente en Europe, envoyez-la nous !

Acheter les programmes américains directement aux États-Unis, cela peut se défendre (pour qui accepte d'envoyer outre-Atlantique son numéro de carte bleue et sa signature !). Mais pour le matériel (cartes mémoire, interfaces, lecteurs de disques, etc.), la notion de service après-vente en France prend tout son sens : il semble assez difficile d'imaginer un renvoi de votre disque dur outre-Atlantique au moindre pépin.

Nos lecteurs ont donc été heureux d'apprendre dans Pom's 34 qu'une société française, la Société **Bréjoux** de Lyon, arborant sur son blason la noble devise «Apple // for ever», assure la distribution exclusive en France des produits du principal fabricant d'accessoires pour la gamme Apple //, Applied Engineering. *«À de rares exceptions près — nous assure-t-elle — les nouveautés (matériels et patches) sont disponibles en France en même temps qu'aux USA à des prix... américains».*

On peut donc trouver en France la meilleure carte d'extension mémoire type Apple pour les Apple ][+, //e et IIGS, la carte **Ramfactor**, la carte Super-Série plus horloge **Serial Pro**, la carte d'interface parallèle **Parallel Pro** avec option buffer possible (GSphiles, n'oubliez pas que vous n'avez pas de port parallèle intégré, mais que votre port série inclut un buffer réglable de 2 à 64Ko), une horloge pour //c, la carte **PC Transporter**, etc. Je rappelle aussi que la carte **Z-Ram** d'**Applied Engineering** est, avec la carte **Multiram CX** de **Checkmate Technology**, le seul moyen d'étendre la mémoire d'un //c 128Ko français.

Deux informations supplémentaires :

- attendez-vous à entendre bientôt parler d'une carte à numériser les images d'**Applied Engineering** ;
- si vous avez une carte **Ramfactor**, lisez l'article de **Computist** de Janvier 88 page 17 avant d'y ajouter le **Ramcharger** pour maintenir la carte alimentée en permanence.

Vous trouvez la résolution de votre //e insuffisante, même en double-haute résolution (560 x 192). Vous avez noté que le GS rajoute plein de couleurs, mais n'étend la résolution qu'à 640 x 200 maximum, ce qui n'est pas beaucoup plus. Le Mac vous attire, mais dépasse votre budget. Mettez donc dans votre //e une résolution de 640 (horizontal) par 768 (vertical) : c'est plus que l'écran du Mac, et ça ne coûte que 400 dollars (sans le moniteur, évidemment). Vous pourrez lire les disquettes Mac avec votre lecteur 3'5 pour en récupérer les dessins. La carte remplace la carte souris du //e, elle ne prend donc pas un slot de plus (sur GS, cette carte fonctionne, mais elle est trop grande pour rentrer sous le capot !). Un très bon soft graphique accompagne le matériel. L'ensemble s'appelle **The Graphics Toolkit**,

et est vendu par **Demco Electronics**.

Savez-vous qu'on peut facilement et rapidement transformer un Unidisk 3'5 (hors garantie) pour //c et //e en lecteur externe pour le Mac ? Il suffit de lui enlever ce qu'il a 'en trop' (à savoir la carte avec microprocesseur intégrée dans le lecteur qui prend le relais de l'unité centrale trop lente) pour transformer ce lecteur *intelligent* en lecteur *bête*. Il reste d'ailleurs pilotable par la carte UDC sur //e. Si des lecteurs souhaitent des précisions, qu'ils nous le fassent savoir.
Les heureux possesseurs de lecteurs Apple 3'5 GS ne sont pas concernés : leurs lecteurs se comportent bêtement ou intelligemment selon la machine.

# Patchworks

Si vous avez une imprimante Epson, vous n'ignorez sans doute pas qu'AppleWorks 1.4 a hérité de son grand frère US AppleWorks 2.0 un bug gênant qui interdit d'entrer le code ASCII 00 (Contrôle-à) dans la définition des codes de *Mon imprimante*. Le patch de **Randy Brandt** dans **Patchmania** étant un peu torturé, je préfère celui de **Scarlett** de Décembre 87. Le voici adapté pour Appleworks 1.4 VF :

```
UNLOCK APLWORKS.SYSTEM ↵
BLOAD APLWORKS.SYSTEM, A$
   2000, TSYS↵
POKE 11768, 250↵
BSAVE APLWORKS.SYSTEM, A$
   2000, T255↵
```

Le patch suivant pour AppleWorks 1.4 (adapté de la même source) est une surprise pour tous ceux qui ont les caractères souris. Essayez-le.

```
UNLOCK APLWORKS.SYSTEM ↵
BLOAD APLWORKS.SYSTEM, A$
   2000, T$FF↵
POKE 11689,64↵
BSAVE APLWORKS.SYSTEM, A$
   2000, TSYS↵
```

NB : si vous avez installé Super-MacroWorks, remplacez *APLWORKS.SYSTEM* par *APLWORKS.SYS* dans les deux patches ci-dessus. Faut-il rappeler qu'on ne patche que des copies, jamais l'original ?

Le patch principal de ce numéro est carrément un programme : le programme *SETUP.SYSTEM* de **Sean Nolan** permet de donner à ProDOS 8 les possibilités d'initialisation de la machine avant le lancement du premier programme qui étaient jusqu'ici réservées à ProDOS 16. Ce programme doit impérativement être le premier fichier système de la disquette portant le nom *xxx.SYSTEM*. Il sera donc automatiquement exécuté au boot par ProDOS.

*SETUP.SYSTEM* commence par chercher sur la disquette un sous-catalogue appelé *SETUPS*. S'il ne le trouve pas, ou si ce sous-catalogue est vide, il passe la main au second fichier système *xxx.SYSTEM* de la disquette. Mais s'il trouve dans ce sous-catalogue des fichiers type BIN ou type SYS, il les exécute successivement dans l'ordre du sous-catalogue, ignorant les fichiers d'un autre type. Ensuite, il repasse la main au fichier *xxx.SYSTEM* suivant du catalogue principal.

Bien sûr, il y a quelques contraintes à respecter : les fichiers type BIN du sous-catalogue *SETUPS* doivent se terminer par un RTS ($60), les fichiers type SYS par un appel au Quit du ProDOS. Bien entendu, les fichiers SYS doivent démarrer en $2000. Toute la place mémoire entre $800 et $B8FF, ainsi que l'espace $200-$3EF, sont utilisables. Les pages $B9 à $BF sont réservées à la page globale de ProDOS et à *SETUP.SYSTEM* lui-même.

Sean Nolan a mis ce programme dans le domaine public, dans l'idée qu'il constitue un STARTUP universel pour ProDOS 8. Ce programme sera particulièrement précieux à tous ceux qui doivent mettre en place un driver de carte horloge, initialiser un disque Ram (Ramworks, Checkmate, carte type Saturn), aux GSistes qui veulent installer leurs CDA au démarrage, etc. Désormais toute disquette ProDOS 8 devrait avoir le fichier SETUP.SYSTEM en premier fichier système, et contenir un sous-catalogue SETUPS. C'est en tout cas ainsi que se présenteront désormais les disquettes ProDOS 8 (3'5 et 5'25) de Pom's. Rappelons à cette occasion que ProDOS 8 et Basic .System n'étant pas, eux, du domaine public, il faut toujours les rajouter vous-même sur ces disquettes.

Le listing étant trop long pour cette rubrique, c'est sur la disquette Pom's que vous trouverez le programme. Le source, modèle de programmation efficace en ProDOS, se trouve dans Call Apple de **Novembre 87**. Utilisez ce programme sans le modifier : c'est un nouveau standard. Ne croyez pas ceux qui vous disent qu'on peut le renommer ATINIT.

# Des bugs dans Pom's ?

Espérons que vous ne verrez pas trop souvent cette nouvelle rubrique dans *Apple // for ever* : elle est destinée à corriger mes propres erreurs, merci aux lecteurs. Consolons-nous : quatre erreurs, quatre bonnes nouvelles.

D'abord à propos de la carte **Speedisk** : pas d'incompatibilité avec le nouveau système du GS, et ProDOS 16 1.3. Les GSistes ne seront donc pas privés de turbo.

Deuxième bonne nouvelle : la nouvelle version de la carte **Universal Disk Controler** de **Central Point Software** est débarrassée de la plupart de ses bugs et ne pose plus les problèmes d'encombrement de la première version. Si vous achetez la carte et un lecteur 3'5, vous avez la dernière version de Copy ][ Plus en prime.

Troisième bonne nouvelle : la Société **Bréjoux** nous informe que la version de la carte **PC Transporter** d'Applied Engineering qu'elle diffuse (pour IIGS seulement pour le moment) est désormais capable de formater et démarrer directement les disquettes 3'5 en MSDos avec le lecteur Apple 3'5. Reste évidemment que 80% au moins des logiciels MSDos sont sur 5'25, il en est de l'IBM comme de l'Apple //. Réponse : Bréjoux fait la conversion pour ses clients. L'achat du lecteur 5'25 MSDos Applied Engineering n'est donc pas indispensable... si les

applications MSDos que vous utilisez ne sont pas protégées.

Enfin à propos de **DDA**, sœur française de **l'APDA** américaine : l'adresse publiée dans Pom's 34 est celle du versant Mac. Le versant IIGS, c'est la société **Prim'vert**. La bonne nouvelle, c'est que le support technique de DDA est assuré pour le IIGS par "la Femme aux Clefs d'Or", j'ai nommé **Nicole Bréaud-Pouliquen**. Réponse assurée les lundis, mardis après-midi et mercredis matin au 37 47 53 21 (province).

## AppleWorks 2.0 et ↵

Dans le numéro 33 de Pom's page 67, Jean-Yves Bourdin proposait un patch pour remplacer le caractère 'damier' un peu rétro par le caractère ↵ plus adapté pour symboliser un retour-chariot. Ce patch était conçu pour la version française 1.4.
Voici le même, adapté à la version U.S. 2.0 :

```
BLOAD SEG.M1,T0,A$2100,L$E00,B$6E00↵
POKE 8929,205↵
POKE 9333,205↵
POKE 11665,205↵
UNLOCK SEG.M1↵
BSAVE SEG.M1,T0,A$2100,L$E00,B$6E00↵
LOCK SEG.M1↵
```

Philippe Ayral

## Encore une partie et j'éteins

Le jeu plante au 27e tableau. Mon vaisseau refuse de décoller. Je ne peux pas sortir des oubliettes. Tu copies les pistes impaires avec EDD3 mode 5, les pistes paires avec Locksmith 4.1 synchro, et si ça plante tu recommences tout. Je n'ai pas la documentation. Je l'ai, mais elle est illisible. J'ai l'impression qu'il me manque une face. Il ne se charge pas sur mon //c... Ah le long et silencieux calvaire du pirate de logiciels de jeux ! Et tout ça pour... 10 dollars !

Eh oui : les plus grands éditeurs de jeux ont aujourd'hui cette politique de vendre à prix bradés leurs jeux un peu anciens, mais de qualité, ceux, en gros, qui doivent faire partie de toute bonne logithèque. 10 à 15 dollars, ça ne met vraiment pas cher le prix de la vertu, du droit de rouspéter, et du confort d'un exemplaire original de : **Summergames 1, Temple of Apshai Trilogy, The Movie Monster Game, World Greatest Baseball** et **Football Games**, d'**Epyx** ; **Choplifter** de **Broderbund** ; **Six Gun Shootout**, de **SSI** ; **Sargon III**, de **Hayden Software** ; **Gato**, de **Spectrum Holobyte** ; **Adventure Construction Set, Age of Adventure, Archon I** et **II, Lords of Conquest, Movie Maker, Murder Party, Music Construction Set, One on One, Pinball Contruction Set, Realm of Impossibility, Seven Cities of Gold, Skyfox, Super Boulder Dash**, tous d'**Electronic Arts**. (Prix relevés dans **Nibble** et **InCider** de Janvier).

Ne croyez pas que je veuille vous dissuader de jouer aux pirates sur Apple // : **Pirates !** de **Micro-Prose** (vous savez, ceux qui ne produisent pas WordStar), combine simulation, jeu de rôles et stratégie. **High Seas**, de **Gardé Games of Distinction**, vous fait commander de magnifiques vaisseaux et mener de très durs combats. Dans **Plundered Hearts** d'**Infocom** jeu d'aventures, en texte uniquement, vous êtes la belle Angélique, prisonnière des pirates. N'hésitez plus : hissez le drapeau noir...

— Allô, Michel ? (Michel consent parfois à lâcher son joystick pour donner quelques brefs avis aux lecteurs de Pom's. Un expert).

— Si vous avez aimé **King Quest** de **Sierra On Line**, vous aimerez plus encore son **Space Quest**, très bon en 8 comme en 16 bits. Par contre, la version 8 bits de **World Games** d'**Epyx** n'est pas du tout à la hauteur de son **California Games**. Jouez-y sur GS. **Thunder Shopper** d'**Actionsoft** est plus maniable comme hélicoptère que Tomahawk de Datasoft. Mais guider cette machine dans les airs est presque aussi difficile que piloter le **Destroyer** d'**Epyx** sur l'eau. Heureusement, il a une option d'entraînement : emportez la disquette au bureau. **Wizardry IV** de **Sir-Tech** est sorti : il faut réellement être devenu un expert dans les trois précédents pour pouvoir y jouer. Alors, pour **Dondra**, très bon début d'une nouvelle série, les **Questmaster**, de **Spectrum Holobyte** pour Apple // et GS, commencez à jouer dès maintenant pour accumuler les points d'expérience.

— Merci Michel, travaille bien. Je te rappelle dans deux mois.

## 16 bits

Maintenant que vous avez les nouvelles Roms, tapez £ sous moniteur : le moniteur lui-même et le Memory Peeker seront installés en CDA. Voici un mini-programme (**Sean Nolan, Call Apple** Février 88) pour ProDOS 8 qui se charge d'installer automatiquement ces deux CDA au démarrage (si vous avez autre chose qu'un GS avec les Roms 01, il ne fait rien). Vous le trouveiez aussi sur la disquette Pom's 35 à sa place : dans le sous-catalogue *SETUPS*.

```
CALL -151↵
*300 : 38 20 1F FE B0 0C
       8A D0 09 C0 01 D0 05
       A0 00 20 84 FE 60↵
*BSAVE MONITOR.INSTALL, A
       $300, L$13 <RETURN>
```

On trouve tout sur GS : vous pouvez

même maintenant utiliser le disque /RAM5 du GS comme disque Ram pour... le DOS 3.3, grâce au programme **RAM 3.3** de **RDC Inc.** Vous pouvez ajouter jusqu'à huit disques Ram de 192Ko. Mais le patch pour le Ramdisque, plus le patch pour les 3'5, plus le patch pour le disque dur, plus vos patches habituels, je ne sais pas si ça fait un patchwork vraiment harmonieux...

Enfin ! Possesseurs de GS et d'une imprimante non-Apple, vous allez enfin pouvoir imprimer vos images GS. La dernière version de **816Paint**, de **Baudville**, reconnaît l'essentiel des imprimantes et des cartes d'interface du marché. Je n'ai rien contre l'ImageWriter, mais une machine ouverte comme le GS doit permettre à l'utilisateur de choisir ses périphériques. Espérons que les autres programmes 16 bits vont s'y mettre vite.

Après la carte **Memory Saver** de **Checkmate Technology**, voici la carte **Ramkeeper** d'**Applied Engéniering** qui, elle aussi, permet de transformer une partie de votre disque Ram en disque Rom auto-alimenté, et de mettre deux cartes mémoire GS dans le même slot (vérifier que l'alimentation de la carte soit prévue pour notre secteur 220V). Supériorité : elle marche aussi avec la carte **GSRam Plus** d'**AE**, ce que ne permet pas pour l'instant **Memory Saver**. Mais attention, avec ces deux cartes, aux compatibilités entre les cartes que vous mettez et à la liberté du slot 7, sans parler des endroits bizarres où vous allez retrouver vos drives du slot 5 quand vous aurez à la fois un Ramdisk et un Romdisk. Si vous cherchez un disque Ram auto-alimenté qui ne mange pas la mémoire GS et ne déplace pas vos drives, songez à la carte **Speedisk**.

À ce propos, certains GSmaniaques gourmands, se trouvant déjà à l'étroit avec 1 Méga, m'ont demandé de faire le point sur les cartes d'extension mémoire du GS. En fait, il y a une telle diversité de cartes et le prix des puces mémoire est si fluctuant (et dans le mauvais sens en ce moment) qu'il faudrait revenir sur la question à chaque numéro de Pom's. Je me limiterai donc aujourd'hui à la liste des cartes et des fabricants (autres qu'Apple) que je connais, et à une liste des questions à ne pas oublier de poser à chacun avant achat.

Fabricants :

**Applied Engineering** (GS Ram, GS Ram Plus)
**Applied Ingenuity** (GS Juice)
**AST** Research (RamStack Plus)
**Checkmate Technology** (Multiram GS)
**MDIdeas** (Octoram)
**On Three** (On Board)
**Orange Micro** (Rampak 4 GS)
**Parallax** (MegaRAM)

Questions :

— Jusqu'à combien cette carte seule peut-elle porter la mémoire du GS ? Et avec carte additionnelle ? Quel est le prix d'une carte additionnelle vide ?

— Quel type de puces mémoire utilise-t-elle ? Combien le même vendeur vend-il aujourd'hui 1 Méga de puces de ce type ?

— Puis-je installer sur cette carte les puces mémoire que j'ai déjà sur d'autres cartes (GS ou //e) ? Puis-je l'acheter avec 0Ko installé ?

— La carte et ses additions laissent-elles le slot 7 entièrement libre ?

— Supporte-t-elle un disque Rom ? De quelle façon ? Taille maximum de ce disque Rom ? Peut-on le protéger contre l'écriture ?

— Supporte-t-elle l'accès mémoire direct (DMA) sans reconfiguration du Ramdisk ?

— Des logiciels sont-ils fournis avec la carte ? Lesquels ? Ces logiciels sont-ils compatibles avec les cartes des autres marques ?

La version 3.0 de **Multiscribe GS** (ProDOS standard dans la version déprotégée) de **Styleware**, qu'espérait Bernard Toméno dans Pom's 33, inclut un correcteur et un dictionnaire de synonymes, un driver d'imprimante ImageWriter amélioré, de nouvelles capacités graphiques. S'y adjoignent également un premier volume de Font Library, un ensemble d'accessoires GS (CDA) appelé **Deskworks** (ProDOS standard). Et Styleware nous promet pour bientôt des drivers d'imprimante Epson pour Multiscribe GS. Il nous promet aussi pour bientôt un éditeur de polices de caractères GS. Peut-être allons-nous prochainement ne plus être contraints d'utiliser Multiscribe //e sur GS (qui en est d'ailleurs également à la version 3.0, la même que l'autre mais avec un correcteur, plutôt lent, en accessoire).

**Word Perfect 2.0 GS** de **Word Perfect** Corporation (ProDOS standard) est maintenant le traitement de textes (non graphique) le plus puissant et le plus complet sur GS. Correcteur et dictionnaire de synonymes, enregistrement des macros en direct, indentation gauche et droite, coupure automatique des mots, tout y est. En plus, interface graphique-souris. Multiscribe ou Word Perfect ? Réponse : rien à faire, il faut les deux.

À propos des ports série intégrés du GS : ce qui était possible pour AppleWriter (cf Patchworks dans Pom's 34), ne l'est en général pas pour les autres programmes qui refusent de marcher avec le port GS. En effet les programmes incompatibles sont ceux qui s'adressent directement aux puces de la carte Super Série ou du //c au lieu de suivre les protocoles d'interface définis par Apple. Je ne vois guère que deux solutions : ou bien se procurer une nouvelle version du programme avec un driver spécial GS (solution que je recommande pour les programmes de communication), ou bien mettre la carte Super Série qu'on a gardé de son //e ou racheté d'occasion à bon marché dans le slot 1, et jouer du tableau de bord avant d'imprimer pour choisir entre SSC et port intégré. La plupart des programmes spécifiques GS acceptent d'ailleurs la Super Série dans les configurations.

Non, ne protestez pas : ce n'est pas une faiblesse, mais une force du GS. Connaissez-vous beaucoup de fabricants d'ordinateurs qui poussent le respect de l'utilisateur jusqu'à se

débrouiller pour garder leurs nouveaux modèles compatibles avec les programmes anciens qui violent volontairement les règles du constructeur ?

## À lire

*«Ce bouquin est introuvable. Je m'en fous, je l'ai.»* (Professeur Choron). Une des conséquences de la fin de la production des Apple //e et //c, c'est qu'un certain nombre de livres indispensables à tout possesseur d'Apple //, GS compris, ne seront pas réédités quand le stock sera épuisé. Si vous ne les avez pas encore, et s'il n'est pas déjà trop tard, pensez à vous procurer d'urgence :

**The Dostalk Scrapbook**, par Tom Weishaar et Bert Kersey, chez Tab Books.

**Beneath Apple DOS**, par Don Worth et Pieter Lechner, chez Quality Software.

**ProDOS, inside and out**, par Dennis Dom et Tom Weishaar, chez Tab Books.

**Beneath Apple ProDOS**, par Don Worth et Pieter Lechner, chez Quality Software.

**ProDOS advanced features for programmers**, par Gary Little, chez Brady Communications-Prentice Hall.

**Inside the Apple //e** et **Inside the Apple //c**, par Gary Little, chez Brady Communications-Prentice Hall.

**Understanding the Apple //e**, par Jim Sather, chez Quality Software.

**Assembly cookbook for the Apple ][ //e**, par Don Lancaster, chez Howard Sams and Co.

**Enhancing your Apple ][ //e, Volumes 1 et 2**, par Don Lancaster, chez Howard Sams and Co.

**All about Applesoft**, par Call Apple.

**All about Pascal**, par Call Apple.

**All about DOS**, par Call Apple.

**Disassembly lines**, Volumes 1 à 4, par Sandy Mossberg, chez Microsparc.

Un bon livre en français sur le GS et le ProDOS 16 : **Système ProDOS de l'Apple II GS**, de Marcel Cottini, chez PSI. Il est absolument sans concurrent pour le moment.

Pour piloter un hélicoptère, Michel nous a dit qu'il fallait s'entraîner au bureau. Mais même pour un petit avion, un instructeur nous serait bien utile. Pour ceux qui n'ont pas d'Apple // au bureau, pensez à mettre dans votre serviette **Flight Simulator Copilot**, de **Charles Gulick**, en français chez **PSI**. Entièrement dédié à l'apprentissage de situations de vol sur **Flight Simulator II** de **Sublogic**, ce livre vous économisera pas mal d'heures de vol réel pour avoir votre brevet.

Si vous écrivez des jeux pour ordinateur, ou envisagez de le faire, ou simplement aimeriez savoir qui les fait et comment, abonnez-vous à la nouvelle et intéressante publication écrite par ceux qui vous font passer tant d'heures passionnantes sur votre Apple // : **The Journal of Computer Game Design**. Il faut beaucoup travailler et beaucoup penser pour amuser les gens.

---

**Activision**
2350 Bayshore Parkway, Mountain View, CA 94043, USA.

**Apple Fest 88 Boston**
Cambridge Marketing Inc, One Forbes Road, Lexington, MA 02173, USA.

**Applied Engineering**
P.O. Box 798, Carrollton, TX 75006, USA.

**Applied Ingenuity**
14992 Ramona Boulevard, Unit M, Baldwin Park, CA 91706, USA.

**Association Valentin Haüy**
5 Rue Duroc, 75007 Paris. ✆ (1) 47 34 07 90.

**Ast Research**
2121 Alton Avenue, Irvine, CA 92714, USA.

**Baudville**
1001 Medical Park Drive S.E., Grands Rapids, MI 49506, USA.

**Beagle Bros/Software Touch**
6215 Ferris Square, Suite 100, San Diego, CA 92121, USA.

**Berkeley SoftWorks**
2150 Shattuck Avenue, Berkeley, CA 94704, USA.

**Big Red Computer Club (BRAC)**
423 Norfolk Avenue, Norfolk, NE 68701, USA.

**Brady Communications, Inc.**
Prentice-Hall Publishing, Bowie, MA 20715, USA.

**Bréjoux**
29 rue Montribloud, 69009 LYON. ✆ 78 36 52 69

**Broderbund Software, Inc.**
17 Paul Drive, San Rafael, CA 94903, USA.

**Call Apple**
290 S.W. 43rd Street, Renton, WA 98055, USA.

**Central Point Software**
9700 S.W. Capitol Highway, £100, Portland, OR 97219, USA.

**Checkmate Technology**
509 South Rockford Drive, Tempe, AZ 85281, USA.

**Claris Corporation**
440 Clyde Avenue, Mountain View, CA 94043.

**Club micro son**
3 Rue Berthe, 92370 Chaville. ✆ (1) 47 50 16 49.

**Computist**
P.O. Box 110846-T, Tacoma, WA 98411, USA.

**Data Pak Software**
14011 Ventura Boulevard, Suite 507, Sherman Oaks, CA 91432, USA.

**Datasoft**
19808 Nordhoff Place, Chatsworth, CA 91311, USA.

**Demco Electronics**
10516 Grevillea Avenue, Inglewood, CA 90304, USA.

**Electronic Arts**
Electronic Arts Direct Sales, P.O. Box 7530, San Mateo, CA 94403, USA.

**Epyx, Inc.**
600 Galveston Drive, Redwood City, CA 94063, USA.

**FWB, Inc.**
2040 Polk Street, Suite 215, San

Francisco, CA 94109, USA.

**Garde Inc.**
8 Bishop Lane, Madison, CT 06443, USA.

**Howard Sams and Co, Inc.**
4300 West 62nd Street, Indianapolis, IA 46268, USA.

**Infocom**
125 Cambridge Park Drive, Cambridge, MA 02140, USA.

**Living Legend Software**
1915 Froude Street, San Diego, CA 92107, USA.

**MDideas**
1163 Triton Drive, Foster City, CA 94404, USA.

**Micol Systems**
9 Lynch Road, Toronto, Ontario, CANADA.

**MicroProse**
180 Lakefront Drive, Hunt Valley, MD 21030, USA.

**Microsparc, Inc.**
52 Domino Drive, Concord, MA 01742, USA.

**On Three**
4478 Market Street, Suites 701-702, Ventura, CA 93003, USA.

**Orange Micro, Inc.**
1400 N. Lakeview Avenue, Anaheim, CA 92807, USA.

**P. Ingénierie/Claris France**
10 Rue Mercoeur, 70011 Paris. Tel (1) 43 70 71 72

**Parallax, Inc.**
5249 Locust Avenue, Carmichael, CA 95608, USA.

**PBI Software, Inc.**
1163 Triton Drive, Foster City, CA 94404, USA.

**Pinpoint Publishing**
5901 Christie Avenue, Emeryville, CA 94608, USA.

**Prim'vert/DDA**
36 Rue des États Généraux, 78000 Versailles.

**PSI**
BP 86, 77402 Lagny sur Marne Cedex.

**Quality Software**
21601 Marilla Street, Chatsworth, CA 91311, USA.

**RDC, Inc.**
408 South Baldwin Street, Madison, WI 53703, USA.

**Roger Wagner Publishing**
1050 Pioneer Way, Suite P, El Cajon, CA 92020, USA.

**Scarlett**
voir Big Red Computer Club

**Sierra on Line**
5750 France Avenue South, Suite 123, Edina, MN 55435, USA.

**Sit-Tech Software, Inc.**
P.O. Box 245, Ogdensburg, NY 13669, USA.

**Spectrum Holobyte**
Sphere Inc, 2061 Challenger Drive, Alameda, CA 94501, USA.

**Strategic Simulations Inc. (SSI)**
1046 N. Rengstorff Avenue, Mountain View, CA 94043, USA.

**StyleWare**
5250 Gulfton, Suite 2E, Houston, TX 77081, USA.

**Sublogic**
713 Edgebrook Drive, Champaign, IL 61820, USA.

**Support System International**
150 South Second Street, Richmond, CA 94804, USA.

**Tab Books, Inc.**
Blue Ridge Summit, PA 17214, USA.

**The Journal of Computer Game Design**
5251 Sierra Road, San Jose, CA 95132, USA.

**Third Wave Technology**
11934 Lorain Avenue, Cleveland, OH 44111, USA.

**TimeWorks**
444 Lake Cook Road, Deerfield, IL 60015, USA.

**Word Perfect Corporation**
288 West Center Street, Orem, UT 84057, USA.

**Zedcor**
4500 E.Speedway, Suite 22, Tucson, AZ 85712, USA

# Courrier des Lecteurs

## Sur Calvacom

### Def Fn modifiable

1) Problème en Basic
De: David BENSIMON (DB25) - 16 fev 88 09h17

Je viens d'acheter un Mac SE et commence à écrire un programme en MSBasic 3.0 interprété, qui permet de tracer des fonctions mathématiques. J'ai un problème : je voudrais que la fonction soit saisie dans le programme et ne sait comment faire. J'ai essayé de faire INPUT A$ (où A$ est la fonction) puis DEF FNF(X)=A$ mais Basic interprète cela comme si ma fonction était une constante chaîne. Pouvez-vous m'aider ?

*Vous soulevez là un problème clasique en Basic. Il n'y a pas de solution simple ; sur l'Apple II, la méthode consiste à créer une ligne REM de 250 caractères quelconques puis de la modifier par programme par des POKEs pour créer le DEF FN de son choix.*
*Sur le Mac, c'est impossible car les blocs de mémoire se déplacent allègrement.*
*Il convient donc... de se pencher sur la question !*

### Longueur de page sur ImageWriter II

2) Impression sur une page 21/29,7
De: Pascal CARRENO (PC94) - 16 fév 88 21h15

Existe-t-il une commande (code ASCII) pour définir la longueur d'une page sur ImageWriter II. La documentation que je possède ne précise rien à ce sujet. Ceci permettrait d'éviter une série de PRINTs pour

terminer une page. Par la même occasion, pouvez-vous m'indiquer les commandes pour 11 et 12 pouces ?

*Sur l'ImageWriter II, pas de problème, il y a une commande simple :*

*ESC H nnnn*

*où nnn est la longueur de la page exprimée en 144ème de pouce. Depuis le Basic, voici les instructions :*

*PRINT CHR$(27)"H1584" pour 11 pouces et*

*PRINT CHR$(27)"H1728" pour 12 pouces.*

*Sur le Macintosh, vous mettriez LPRINT à la place de PRINT.*

*En ce qui concerne l'ImageWriter I, plutôt qu'un long discours, voici un programme qui indique à l'imprimante que son papier comporte NL lignes par page :*

```
10 PRINT D$"PR£1
20 NL = 66 : REM 66 = 11 pouces, 72 = 12 pouces
30 PRINT CHR$(29) CHR$(65) CHR$(64);
40 FOR I = 1 TO NL: PRINT CHR$(64) CHR$(64);:
   NEXT
50 PRINT CHR$(67) CHR$(64) CHR$(65) CHR$(64)
   CHR$(30)
60 PRINT D$"PR£0
```

*NB: n'oubliez pas les ';' des lignes 30 et 40. Une page comporte au maximum 96 lignes.*

*Après l'envoi de ces codes, l'imprimante saute le nombre de lignes qui convient pour rejoindre le haut de page suivant dès réception d'un CHR$(12).*

*Certains codes de contrôle de l'ImageWriter nécessitent 8 bits significatifs, autrement dit 0 n'est pas toujours équivalent à 128 ; or le Basic envoie toujours le bit de poids fort à 1 (toujours 128 à la place de 0).*
*Dans Pom's 34, article Initiation, une routine résoud cette difficulté.*

## Du type BAS au type TXT

6) question à domicile... *AR*
De: Eric VANTROEYEN (EV17) - 21 fev 88 10h16

J'aimerais savoir comment faire pour récupérer avec un traitement de textes un programme Basic (mise en format TXT).
Comment peut-on simuler la touche FNCT du Minitel (code ASCII) pour le faire passer à 4800 bauds ? Ceci permettrait de définir 'Mon Imprimante' pour qu'AppleWorks imprime sur le Minitel...

*Il suffit d'ajouter à votre programme :*

```
60000 D$= CHR$(4)
60010 PRINT D$"OPENFICHIER.TEXT
60020 PRINT D$"WRITEFICHIER.TEXT
60030 LIST,59999
60040 PRINT D$"CLOSE
```

*puis de faire : RUN 60000.*

*Pour faire passer le Minitel à 4800 bauds au niveau de la prise, il faut envoyer les codes suivants :*
*$1B, $3A, $6B et un octet de vitesse constitué ainsi :*

```
bit 6 = 1
bit 3 à 5 = vitesse d'émission
bit 0 à 2 = vitesse de réception
   001 = 75 bauds
   010 = 300 bauds
   100 = 1200 bauds
   110 = 4800 bauds
```

*Donc pour 4800 bauds, l'octet de vitesse vaut : 1110110 soit $76, soit encore 118.*

*Il convient de préciser que ceci ne modifie en rien la vitesse de transmission sur le réseau téléphonique.*

## Répom'deur et Hgr.Mntl

11) Repomdeur
De: Philippe CUVELETTE - 21 fev 88 23h48

Je me suis inspiré de ENVOIIM et j'ai créé une première page à mon serveur. Il s'agit d'un fichier TXT créé avec HGR.MNTL (Pom's 33). La ligne permettant de l'envoyer à mon correspondant éventuel est : GET R$: & AFFICHE,R$;
Cela envoi une image brouillée.
Si la même ligne est GET R$: PRINT R$; tout devient normal. Pourquoi ?

*Recodage, recodage. La routine AFFICHE effectue toujours un recodage pour éviter les problèmes dus aux couples £#, §], °[, à@, etc. Bien sûr cela pose un problème pour le graphisme. Dans Pom_Link 3.0, une routine envoi brutalement, sans aucun recodage et le problème est résolu. Ceci est aussi valable pour la version Macintosh de Pom_Link.*

## Pom's 27 & système Mac

1) CABLE COM MAC+
De: Arian ZELWER (AZ10) - 06 jan 88 09h41

Votre câble de liaison semble poser des problèmes avec mon MacPlus et 'Minitel1' (Pom's 27). Le clavier est décalé : W = <, X = W, C = X etc. Je n'ai plus de Return !

*Non, n'accusez pas le câble : le problème vient de la disquette 27 de Pom's qui contient un ancien système, ceci pour permettre le fonctionnement avec les Macintosh 128Ko, encore nombreux à cette époque. Le remède est simple : copiez les programmes Minitel1 et Minitel2 sur l'une de vos disquettes dotées de votre système habituel ou bien copiez votre système sur la disquette 27.*

## Schémateur (Pom's 31) & souris

20) SCHEMATEUR ET SOURIS
De: Pierre DIEUMEGARD (PD11) - 29 dec 87 15h38

Enlever la souris du //c pour y installer une manette de jeu n'est pas pratique et est moins précis. SCHEMATEUR peut fonctionner avec la souris moyennant la modification suivante :

```
25 PRINT CHR$(4)"PR£4": PRINT CHR$(1)
26 PRINT CHR$(4)"PR£0
100 REM SOURIS
101 IF TQ THEN GOSUB 295
102 PRINT CHR$(4)"IN£4":INPUT "",X%,Y%,SB
103 IF X% > 279 THEN X% = 279
104 IF Y% > 191 THEN Y% = 191
106 SB = ABS(SB): IF SB = 1 OR SB = 2 THEN
   XDRAW CU AT X%,Y%
290 POP:TQ=1: RETURN
295 PRINT CHR$(4)"PR£4":PRINT CHR$(1)
296 PRINT CHR$(4)"PR£0": TQ=0
297 RETURN
```

De plus, il faut commencer la ligne 250 par :

```
250 FOR I = 1 TO ZC: IF R=R(I) THEN PRINT
   CHR$(4)"IN£0": ON I GOSUB 290,  310...
```

*Un seul mot : merci*

## Text aléatoire/Text séquentiel

23) FICHIER TEXT
De: Philippe CUVELETTE (PC121) - 29 dec 87 18h56

Comment transformer l'accès d'un fichier T direct pour pouvoir l'utiliser avec un traitement de textes ?

*Si les codes 0 qui complètent les zones d'un fichier aléatoire gênent votre traitement de textes, il s'agit peut-être d'AppleWriter ?*
*Il suffit de lire le fichier direct et de l'écrire dans un fichier séquentiel avec un programme du type :*

```
OPEN fichier direct, Lxxx
OPEN fichier séquentiel
FOR I = 0 TO nombre d'enregistrement
   READ fichier direct, enregistrement I
   INPUT chacun des champs
   WRITE fichier séquentiel chacun des champs
NEXT
```

# Sur le serveur Pom's (1) 39 53 04 40

## Des fleurs

Merci pour vos très nombreux messages sur notre serveur. Dans le premier mois de connexion, la grande majorité a été constituée de félicitations et encouragements.
Pom_Link répond effectivement à un besoin d'ouverture de nos machines : un vrai Basic télématique qui nous semble bien sans bug.

Pardonnez-nous de ne pas avoir toujours accusé réception personnellement mais soyez remerciés ici par la rédaction.
Le serveur est à votre disposition 24 heures sur 24 mais le numéro de téléphone que vous nous indiquez doit, si possible, être accessible aux heures de bureau...

Quelques messages juste pour notre plaisir :

28/02/88 * 16h18
Jean Claude Hazera
Bravo pour tous les programmes de communication et en particulier pour Répomdeur. Bravo et merci.

26/02/88 * 11h05
Patrick Forien
Très bien votre répondeur. Quelle surprise nous réservez-vous dans le numéro de mars . Amitiés.

25/02/88 * 00h07
Hacker Prime
Enfin un vrai Basic télématique pour Apple. Cela ouvre d'étonnantes possibilités pour de futurs serveurs télématiques.

24/02/88 * 00h04
Un Somnanbule
Un grand bravo pour ce superbe soft et encore une ovation pour toute l'équipe de Pom's. Continuez sur cette voie.

22/02/88 * 23h58
François Muller
Bravo pour ce programme, vous souhaite une bonne continuation.

20/02/88 * 14h23
Patrice Triquet
Encore bravo pour votre revue qui reste la seule qui traite de l'Apple // clairement et avec convivialité !
...

## ProDOS & Integer Basic

03/03/88 * 14h52
Roger Riéra

Est-il possible de charger des programmes en Basic Entier avec ProDOS ?

*Hélas non, il vous faut choisir ProDOS ou Integer Basic : il n'y a pas la place pour tout le monde dans la carte langage !*

## Carte Transporter

20/02/88 * 23h45
Jean-Louis Aspirot

Pouvez-vous aider un de vos abonnés à se procurer la carte Transporter de chez Applied Engineering, le kit, le duo disk. Faut-il passer une commande aux USA ?

*Une seule adresse : Bréjoux (coordonnées à la fin de la rubrique Apple // for ever).*

## SuperMacroWorks & AppleWorks 1.2

20/02/88 * 11h47
Georges Cans

Avec AppleWorks 1.2, peut-on se servir de SuperMacroWorks ? Sinon, que faire ?

*SuperMacroWorks nécessite la version française 1.4 ou la version américaine 2.0. MacroWorks nécessitait déjà la version US 1.3.*
*Seul remède donc, consulter votre revendeur ou P-Ingénierie (coordonnées à la fin de la rubrique Apple // forever).*

## Banques de données U.S.

18/02/88 * 21h36
Jean-Pierre Fournier

Nouvelle orientation télématique super. Lequel parmi les logiciels Pom's permet de se connecter aux banques de données U.S. ?

*CLV_Pom's étant conçu pour enregistrer tout serveur ASCII, devrait vous rendre service. La version Apple // fonctionne uniquement avec un Minitel.*

## SuperMacroWorks et AppleWorks 2.0 US

01/03/88 * 00h00
Philippe Ayral

SuperMacroWorks fonctionne-t-il avec AppleWorks 2.0 US?

*Nous serons brefs : oui*

## T_Pom's : adresses à l'écran

28/02/88 * 16h43
Philippe Kachaner

Au sujet de T_Pom's (n° 30), quel programme en langage machine dois-je ajouter pour pouvoir afficher le contenu des fichiers téléphone à l'écran.

*T_Pom's crée des fichiers de type TXT (TEXT/MACA sur le Macintosh) qu'il est simple d'ouvrir depuis tout traitement de textes. Depuis le Basic, on peut simplement réaliser un programme de lecture (OPEN fichier, READ fichier, GET r$, PRINT ...).*

*Pour une question d'occupation mémoire, vous ne pourrez pas simplement ajouter une fonction d'affichage à l'écran*

*mais vous pouvez transformer l'option impression : dans le source page 59, remplacez les deux JSR $C100 par des JSR $FDED. Les caractères qui étaient dirigés vers l'interface le seront vers l'écran.*

## Répom'deur : un petit bug

06/03/88 * 13h54
Claude Alison

Merci pour Répomdeur Pom's 34. Une remarque : les coordonnées du dernier correspondant restent en mémoire ; ce bug sera éliminé par :

```
41 N$="":PR$="":TL$="":PO=0
```

*Effectivement, sur la version Apple // la remise à 0 des variables avait été ajoutée par erreur à la ligne 11 au lieu de la ligne 40.*

## SuperMacroWorks : documentation

27/02/88 * 16h55
Alain Desnoes

Comment puis-je avoir la documentation française de SuperMacroWorks ?

*Si SuperMacroWorks vous a été livré par nos soins, il était accompagné de la documentation française, de la version entièrement francisée du logiciel, et de la disquette Bonus.*

*Si vous l'avez obtenu chez un revendeur, Dimitri Geystor – Lachenaz – 74350 Cruseilles vous l'adressera contre 65,00 F et votre 'Proof of purchase'. Notons au passage que si vous avez acheté ce logiciel en France, le distributeur est légalement tenu de fournir le mode d'emploi en français.*

*Et si vous n'avez pas cette preuve d'achat, il n'y a pas de solution.*

## Minitel, Pom's 27

26/02/88 * 17h08
Jean-François Lebourg

Je vais vous acheter MinBas (disquette Pom's 27) mais je dois poser deux questions :

1- Que signifie 'Restitution hors réseau' ?
2- Est-ce que l'enregistrement s'effectue instantanément ou est-ce qu'il se fait au rythme (lent) de l'affichage du programme Minitel ?

*MinBas, programme publié dans Pom's 27, enregistre en mémoire puis sur disquette vos séquences de consultation Minitel. Ne perdez pas de temps (coûteux en 3615) à lire les écrans à ce moment. Après avoir déconnecté votre Minitel (donc après votre communication téléphonique), demandez à votre ordinateur de vous 'rediffuser' la séquence et là, prenez votre temps, c'est gratuit.*

*L'enregistrement ne se fait pas instantanément mais au rythme de 1200 bauds (150 caractères/seconde). Il n'y a aucun moyen d'accélérer le débit du serveur car il y a une contrainte technique au niveau des liaisons téléphoniques.*

## Temps d'accès sur Speedisk

ClaudeRozé

Est-il possible d'accélérer le chargement de fichiers de type text. Sur Unidisk 3'5, le programme ci-joint charge les deux fichiers en 34 secondes et sur Speedisk en 27 secondes. Est-ce que le Basic limite définitivement la vitesse ?

*Vous vous heurtez au problème du traitement des chaînes en Basic ; la carte Speedisk ne peut rien contre cela. Votre programme charge en effet dans le premier fichier 250 x 2 variables et 146 x 8 dans le second. Au total 1668 variables alphanumériques.*

*Seule solution pour votre programme ProDOS, le compilateur Beagle avec lequel les temps relevés sont les suivants :*

*Unidisk 3'5 : 22 secondes*
*Speedisk : 15 secondes*

*Il semble difficile de passer au-dessous de ces chiffres sans refondre entièrement le programme Basic.*

## Répom'deur, Vs.Tel & Sectrad

28/02/88 * 12h19
Marc Tixière

Votre répondeur est très beau mais il ne marche pas avec VS.TEL et un SECTRAD... Dommage.

*Effectivement, d'ailleurs il ne fonctionne pas non plus avec MacWrite et un fer à repasser. Plus sérieusement, on pourra toujours trouver tel ou tel matériel incompatible avec tel ou tel programme ; reconnaissons que le Minitel étant très largement répandu (3 500 000 unités en circulation aujourd'hui), étant particulièrement fiable et efficace, et – dernier détail – gratuit, notre serveur a un bel avenir...*

*Quant à la compatibilité avec VS.TEL, nous ne voyons pas d'intérêt.*

## Répondeur enregistreur ou détecteur ?

06/03/88 * 12h43
M. Doret

Un répondeur enregistreur peut-il remplacer un détecteur d'appel ?

*Oui, mais... Les répondeurs sont évidemment équipés de détecteurs d'appels ; on peut toujours ouvrir et faire intervenir le fer à souder, mais peut-être est-il préférable de garder sa garantie et de faire un détecteur autonome...*

## Programmes à deux vitesses

02/03/88 * 19h16
Pierre-Alain Buino

Bravo pour votre revue (me suis abonné pdt AppleExpo). Une remarque cependant : il ne faudrait pas que se généralise le programme dans la revue et le super programme à part et payant !!!!
Cela fait en effet 3 fois je crois, où les évolutions de programmes de la revue sont payants... Attention ! la revue risque de baisser en qualité...

*Quelle est la revue qui à proposé à 45 F un véritable Basic télématique avec INPUT contrôlé et autant de fonctions ?*

*Quelle est la revue qui a proposé à 40 F un programme de transfert de fichiers via Minitel au protocole 100 % fiable et qui recrée les fichiers à l'arrivée avec TOUS leurs attributs ?*

*Pom_Link version 3.1 incluant le mode téléinformatique présentant des instructions plus professionnelles' ne ferait certainement qu'encombrer la revue. Même remarque pour InterPom's V2.0 qui, concernant les transferts avec IBM PC et transferts locaux rapides.*

*Pour reprendre votre formule nous dirons : le super-programme dans la revue et le programme spécialisé à part !*

→ *Suite de la page 45*

## Adresses

**Apple Computer France**
Avenue de l'Océanie
Z.A. de Courtabœuf – B.P. 131
91944 Les Ulis Cedex
☎ (1) 69.28.01.39

**Alpha Systèmes**
18, avenue Alsace-Lorraine
38000 Grenoble
☎ 76.43.19.97

**Belden Electronics S.A.R.L.**
69, rue de la belle Étoile
B.P. 50026 – Paris Nord 2
95946 Roissy – Charles de Gaulle
☎ (1) 48.63.25.80

**Claris/P-Ingénierie**
10, rue Mercœur
75011 Paris
☎ (1) 43.70.71.72

**Computer Unlimited**
2 Tbe Business Centre
Colindeep Lane – Colindale
London NW9 6DU
☎ 01-200 8282

**Crex technology**
34, rue Poncelet
75017 Paris
☎ (1) 42.67.80.46

**Honeywell S.A.**
4, avenue Ampère – B.P. 37
78391 Bois d'Arcy Cedex
☎ (1) 30.58.81.12

**Intellia**
10, rue Mercœur
75011 Paris
☎ (1) 43.70.89.40

**Irwin Magnetics**
Regensbergstrasse 89
CH-8050 Zurich
☎ 01/312 70 30

**Kangourou services**
Rue des Sables
ZAC de Pulnoy-Essey
54420 Pulnoy
☎ 83.21.25.33

**MCI**
8, rue de l'Isly
75008 Paris
☎ (1) 42.94.27.67

**Sensible software, Inc.**
335 E. Big Beaver, suite 207, Troy
MI 48083, (313) 528
☎ (313) 528-1950

# Disquettes Pom's Mac A... I

Ces disquettes contiennent des programmes ou documents 'domaine public' ou 'Freeware' : elles sont le résultat d'une sélection conjointe de Pom's et du Club Apple parmi les 25 disquettes initialement proposées par le Club Apple. Cette réduction de plus de 25 disquettes en 9 correspond à une élimination de nombreux programmes qui soit ne fonctionnaient pas sur Macintosh Plus, soit présentaient un intérêt pratique réduit. Vous trouverez ci-dessous un 'catalogue' des disquettes.

### ☞ Disquette Mac A – Divers

Sort Menu, Paint Mover, Mock, Convert DAM, FEdit, Cube, Idle, MultiScrap, C Check, Hex Calc, Megaroïd, Mem Window

### ☞ Disquette Mac B – Polices de caractères

Andover 12 – APL 12 – ASCII 12 – Bookman* 14/24 – Boxes 9/10/12 – Broadway 24 – Century 18/24 Chancery 24 – Chicago by night 12 – Circus 12/18/24/36 – Cirth 48 – Cursive 12 – Cyril 12 East Orange 12 – Elite 12 – Elvish 12 – Eon 12 – Exeter 13 – Greek 9/10/12/14/18/24 Hollywood 12/24 – Hood River 12 – Lineal 18 – Mars 18 – London 18/36 – Long Island 12/18

### ☞ Disquette Mac C – Polices de caractères

N Helvetica Narrow* 14 – Ophir 12/24 – Palo Alto 9/12/18/24/36 – Park Avenue 18 Philly 9/10/12/18/20/24 – Pica 12 – Princeton 12/24 – Ravenna 12/24 – Runes 12/24 – Santiago 12 Silicon Valley 12 – Stencil 12/24 – Symbol* 9/12/18/24 – Tiny 12 – Toronto fixe 9 – Vectors 9/12 Zapf Dingbats* 18

** les polices marquées d'une étoile ne sont pas celles de la 'LaserWriter' portant les mêmes noms.*

### ☞ Disquette Mac D – Accessoires de bureau

ArtThief, Ascii.DA, FKEY (ASCins, ASClip, BigCursor, Clippy, ComKey, DA Key, FadeKey, PanicKey, ScreenToClip, SetSound, ShowKey, Sleep), Poor Boy's, Dec Maker, file Tools, Maxwell, TheBox

### ☞ Disquette Mac E – Applications

Amazing, Analog Clock, Backgammon, Banner, Bricks, Canfiels, DiskUtil, Fast Formatter, MouliMac

### ☞ Disquette Mac F – Applications

DataFlow, Reversi, yapu, Enigma, F/DA Sorter, File Diddler, HexPuzzle, Iago

### ☞ Disquette Mac G – Applications

ResEdit, Icon Editor, JClock, Life, Missile Command

### ☞ Disquette Mac H – Applications

MacBILLBOARD, MacWait, MenuEdit, PackIt, Social Climber, Solitaire, WaveMaker

### ☞ Disquette Mac I – Applications

Snooker, SystemVersion, Adventure of Snake, ThinkTank to WP

Chaque disquette : 80,00 F TTC franco, bon de commande page 71.

Pour les envois par avion, ajoutez 15 F par numéro et/ou par disquette soit, par exemple, 90 F pour un abonnement avec ou sans disquettes.

# Bon de commande

Gagnez du temps ! Avec votre Minitel et de votre carte de crédit, appelez notre serveur gratuit 24 heures sur 24 au (1) 39 53 04 40.

## ➡ Logiciels pour Apple ][

| | | | | | |
|---|---|---|---|---|---|
| **BananaSoft** 140Ko | 200,00 F ☐ | **Édit. Vidéotex** 140Ko | 200,00 F ☐ | **Ludologic** 140Ko | 80,00 F ☐ |
| **Ordico** 140Ko | 200,00 F ☐ | **Dominos** 140Ko | 80,00 F ☐ | **COGO** (src) 140Ko | 200,00 F ☐ |
| **E.P.E 5.1** 140Ko | 200,00 F ☐ | **E.P.E5.1** 800Ko | 200,00 F ☐ | **MaxMoniteur** 140 Ko | 150,00 F ☐ |
| **Clv_Pom's** 140Ko | 200,00 F ☐ | **Clv_Pom's** 800Ko | 200,00 F ☐ | | |
| **InterPom's 2.0** 140Ko | 450,00 F ☐ | **InterPom's 2.0** 800Ko | 450,00 F ☐ | | |
| **Pom_Link 3.1** 140Ko | 450,00 F ☐* | **Pom_Link 3.1** 800Ko | 450,00 F ☐* *(avec carte Joker : 400,00 F)* | | |

**SuperMacroWorks** 500,00 F ☐* *(avec carte Joker : 450,00 F)*
**Compilateur Beagle** 750,00 F ☐* *(avec carte Joker : 675,00 F)*
**Big U** 450,00 F ☐* *(avec carte Joker : 400,00 F)*

## ➡ Logiciels pour Macintosh

| | | | | | |
|---|---|---|---|---|---|
| **Excel efficace** 400Ko | 190,00 F ☐ | **Excel efficace** 800Ko | 175,00 F ☐ | **MacAstuces** | 200,00 F ☐ |
| **Clv_Pom's** | 200,00 F ☐ | **InterPom's2.0** | 450,00 F ☐ | **Raccourci** | 200,00 F ☐ |
| **Pom_Link 3.1** | 450,00 F ☐* *(avec carte Joker : 400,00 F)* | | | | |

## ➡ Logiciels pour IBM PC

**Secrets de Multiplan** 175,00 F ☐ **InterPom's 2.0** 450,00 F ☐

## ➡ Disquettes 'domaine public' pour Macintosh

**Mac 'A'** ☐ **'B'** ☐ **'C'** ☐ **'D'** ☐ **'E'** ☐ **'F'** ☐ **'G'** ☐ **'H'** ☐ **'I'** ☐ 80,00 F par disquette

## ➡ Reliures

**Reliures toilées** pour 6 numéros de Pom's (un an) : _____ exemplaire(s) à 60,00 F, soit _____ F

## ➡ Abonnements

L'abonnement comprend l'attribution de la carte Joker qui offre un accès privilégié à notre Hotline (assistance dans la mise en œuvre des programmes diffusés par Pom's) et qui donne droit à des remises sur les produits Pom's.

**Abonnements pour six numéros à partir du _____, à :**

| | |
|---|---|
| la **revue** Pom's seule | 225,00 F ☐ |
| la **revue** et les **disquettes Apple ][** 140K | 525,00 F ☐ |
| la **revue** et les **disquettes Apple ][** 800K | 625,00 F ☐ |
| la **revue** et les **disquettes Macintosh** | 625,00 F ☐ |
| la **revue** Pom's, les **disquettes Apple ][** 140Ko – 5' 1/4 et les **disquettes Macintosh** | 925,00 F ☐ |
| la **revue** Pom's, les **disquettes Apple ][** 800Ko – 3' 1/2 et les **disquettes Macintosh** | 1025,00 F ☐ |

Envoyez ce bon et votre règlement à : **Éditions MEV – 12, rue d'Anjou – 78000 Versailles**

**Nom :** _____

**Adresse :** _____

**Règlement par :** CB/Visa/Euro/MasterCard ☐ Chèque bancaire ☐ Chèque postal ☐ Mandat ☐

numéro de la carte ____ ____ ____ ____ date d'expiration __ __ __

Montant ____,__ F Signature :

Pour les envois par avion, ajoutez 15 F par numéro et/ou par disquette

# Bon de commande

Gagnez du temps ! Avec votre Minitel et de votre carte de crédit, appelez notre serveur gratuit 24 heures sur 24 au (1) 39 53 04 40.

## ➡ Revues Pom's

| | | | | |
|---|---|---|---|---|
| n° 8 35,00 F ☐ | n° 10 40,00 F ☐ | n° 11 40,00 F ☐ | n° 12 40,00 F ☐ | n° 13 40,00 F ☐ |
| n° 14 40,00 F ☐ | n° 15 40,00 F ☐ | n° 16 40,00 F ☐ | n° 17 40,00 F ☐ | n° 18 40,00 F ☐ |
| n° 19 40,00 F ☐ | n° 20 40,00 F ☐ | n° 21 40,00 F ☐ | n° 22 40,00 F ☐ | n° 23 40,00 F ☐ |
| n° 24 40,00 F ☐ | n° 25 40,00 F ☐ | n° 26 40,00 F ☐ | n° 27 45,00 F ☐ | n° 28 45,00 F ☐ |
| n° 29 45,00 F ☐ | n° 30 45,00 F ☐ | n° 31 45,00 F ☐ | n° 32 45,00 F ☐ | n° 33 45,00 F ☐ |
| n° 34 45,00 F ☐ | n° 35 45,00 F ☐ | | | |

## ➡ Disquettes Pom's Apple ][, 140Ko – 5,25 pouces

Ces disquettes regroupent l'ensemble des programmes pour Apple // d'une revue.
Jusqu'au numéro 28, elles sont au format DOS 3.3.
Depuis le n° 29, le recto est en DOS 3.3, le verso en ProDOS.

| | | | | |
|---|---|---|---|---|
| n° 1+2 60,00 F ☐ | n° 3 60,00 F ☐ | n° 4 60,00 F ☐ | n° 5 60,00 F ☐ | n° 6 60,00 F ☐ |
| n° 7 60,00 F ☐ | n° 8 60,00 F ☐ | n° 9 60,00 F ☐ | n° 10 60,00 F ☐ | n° 11 60,00 F ☐ |
| n° 12 60,00 F ☐ | n° 13 60,00 F ☐ | n° 14 60,00 F ☐ | n° 15 60,00 F ☐ | n° 16 60,00 F ☐ |
| n° 17 60,00 F ☐ | n° 18 60,00 F ☐ | n° 19 60,00 F ☐ | n° 20 60,00 F ☐ | n° 21 60,00 F ☐ |
| n° 22 60,00 F ☐ | n° 23 60,00 F ☐ | n° 24 60,00 F ☐ | n° 25 60,00 F ☐ | n° 26 60,00 F ☐ |
| n° 27 60,00 F ☐ | n° 28 60,00 F ☐ | n° 29 60,00 F ☐ | n° 30 60,00 F ☐ | n° 31 60,00 F ☐ |
| n° 32 60,00 F ☐ | n° 33 60,00 F ☐ | n° 34 60,00 F ☐ | n° 35 60,00 F ☐ | |

## ➡ Disquettes Pom's Apple ][, 800Ko – 3,5 pouces

Ces disquettes regroupent l'ensemble des programmes pour Apple // d'une revue.
Elles sont au format ProDOS.

| | | | | |
|---|---|---|---|---|
| n° 29 80,00 F ☐ | n° 30 80,00 F ☐ | n° 31 80,00 F ☐ | n° 32 80,00 F ☐ | n° 33 80,00 F ☐ |
| n° 34 80,00 F ☐ | n° 35 80,00 F ☐ | | | |

## ➡ Disquettes Pom's pour Macintosh

Ces disquettes regroupent l'ensemble des programmes pour Macintosh d'une revue.

| | | | | |
|---|---|---|---|---|
| n° 14+15+16 150,00 F ☐ | n° 17 80,00 F ☐ | n° 18 80,00 F ☐ | n° 19 80,00 F ☐ | |
| n° 20 80,00 F ☐ | n° 21 80,00 F ☐ | n° 22 80,00 F ☐ | n° 23 80,00 F ☐ | n° 24 80,00 F ☐ |
| n° 25 80,00 F ☐ | n° 26 80,00 F ☐ | n° 27 80,00 F ☐ | n° 28 80,00 F ☐ | n° 29 80,00 F ☐ |
| n° 30 80,00 F ☐ | n° 31 80,00 F ☐ | n° 32 80,00 F ☐ | n° 33 80,00 F ☐ | n° 34 80,00 F ☐ |
| n° 35 80,00 F ☐ | | | | |

## ➡ Recueils de la revue Pom's

Ces recueils regroupent quatre numéros de Pom's.

| | | |
|---|---|---|
| n° 1 (revues 1 à 4) 140,00 F ☐ | n° 2 (revues 5 à 8) 140,00 F ☐ | n° 3 (revues 9 à 12) 140,00 F ☐ |
| Disquettes 1 à 4 200,00 F ☐ | Disquettes 5 à 8 200,00 F ☐ | Disquettes 9 à 12 200,00 F ☐ |

Envoyez ce bon et votre règlement à : **Éditions MEV – 12, rue d'Anjou – 78000 Versailles**

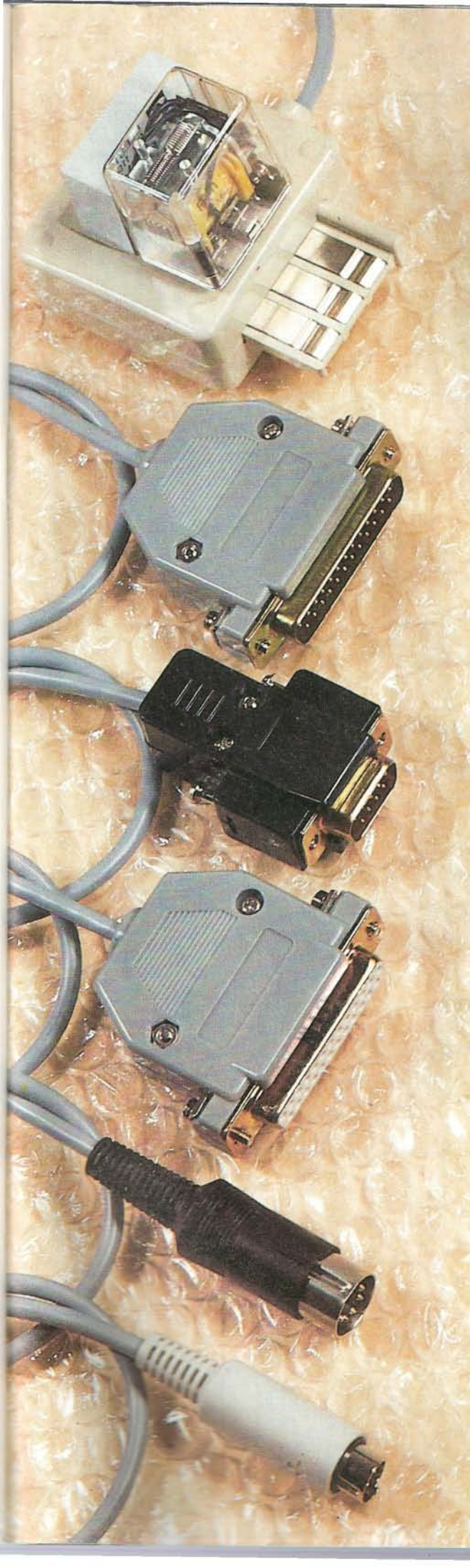
**Nom :** _____

**Adresse :** _____

**Règlement par:** CB/Visa/Euro/MasterCard ☐ Chèque bancaire ☐ Chèque postal ☐ Mandat ☐

numéro de la carte ____ ____ ____ ____ date d'expiration __ __ __

Montant ____,__ F Signature :



## Détecteur d'appels téléphoniques

Cet apparell, pour Apple // ou Macintosh, autorise une surveillance de la ligne téléphonique pour l'utilisation de l'Apple comme serveur avec un logiciel tel, par exemple, Répom'deur publié dans le numéro 34 de Pom's.

## Câble-interface Apple → Minitel

Pour faire fonctionner les programmes suivants :

- *Minitel/1* pour Macintosh, *MinBas* pour Apple ][+, //e, //e+, //c et IIGS : programme permettant l'enregistrement des écrans Minitel, la restitution à loisir hors réseau, le stockage et/ou l'impression de copies d'écran du Minitel, et aussi l'envoi de textes ou messages sur un serveur. Programme du numéro 27 de Pom's.
- *InterPom's 1.0* (et plus) pour Apple ][+, //e, //e+, //c, IIGS et Macintosh : programme de téléchargement entre Apple ][ et/ou Apple ][ et Macintosh. Transmission de n'importe quel type de fichier (système, texte, binaire, Basic...) en utilisant le Modem du Minitel, Version 1.0 publiée dans le numéro 28 de Pom's.
- *T_Pom's* pour Apple ][+, //e, //e+, //c, IIGS* et Macintosh : récupération de l'annuaire téléphonique sous la forme de fichiers texte. Numéro 30 de Pom's.
- *Clv_Pom's* pour Apple ][+, //e, //e+, //c et IIGS* et Macintosh : programmes de communication pour CalvaCom et serveurs 'ASCII'. Numéro 31 de Pom's.
- *Paint → Minitel* pour Mac et *HGR → Minitel* pour Apple ][+, //e, //e+, //c, IGS* : graphisme et Minitel, programmes proposés dans le numéro 33 de Pom's.
- *Répom'deur* pour Macintosh, Apple ][+, //e, //e+, //c et IIGS* : répondeur/enregistreur télématique interrogeable à distance publié dans le numéro 34.

** sur un Apple IIGS, ce programme fonctionne indifféremment avec le port série intégré ou la carte Super Série Apple. Pour connecter le port intégré du IIGS, utilisez un câble pour Macintosh Plus.*

---

### Je désire recevoir :

| | | | |
|---|---|---|---|
| détecteur d'appels Apple // | _____ | à 500,00 F | _____* |
| détecteur d'appels Macintosh | _____ | à 500,00 F | _____* |
| câble Minitel/Apple // & SSC | _____ | à 225,00 F | _____ |
| câble Minitel/Apple //c | _____ | à 225,00 F | _____ |
| câble Minitel/Mac 128, 512K | _____ | à 225,00 F | _____ |
| câble Minitel/Mac Plus, IIGS | _____ | à 225,00 F | _____ |
| câble Minitel/IBM PC™ | _____ | à 225,00 F | _____ |
| câble de liaison locale** | _____ | à 225,00 F | _____ |

** si vous êtes abonnés, vous bénéficiez d'une remise de 10% sur le prix du détecteur, soit 450 F au lieu de 500 F.*

*** préciser le type des deux machines à relier : Mac 512, Mac Plus, Apple //e, //c, IIGS, IBM PC™ .*

*Envois par avion : ajoutez 15 F par câble et/ou détecteur*

---

Éditions MEV – 12, rue d'Anjou – 78000 Versailles

Nom : _____

Adresse : _____

_____

En cas de règlement par CB/Visa/Eurocard/Mastercard :

numéro de la carte _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

date d'expiration _ _ _ _ _ _

Montant _ _ _ _ , _ _ F Signature :

# Petites annonces

Ces petites annonces sont gratuites et réservées aux abonnés (indiquer le numéro de carte Joker). Elles doivent bien entendu concerner l'informatique. Pour les ventes de logiciels, l'annonceur doit nous faire parvenir une photocopie de la facture d'achat.

---

**Vends** Modem Sectrad universel : 1000 F + logiciel d'origine — Version Tel : 300 F — Sideways : 200 F — Unimate : 300 F — La gamme PFS : 1000 F — Magicalc (compatible PFS et extension mémoire) : 500 F.

Jean Ollion - Glénat - 15150 Laroquebrou ✆ 71 62 26 35

---

**Vends** Epson MX/82FT + interface graphique 8133, compatible AppleWorks : 2400 F — Extension mémoire Apple IIGS 256Ko extensible à 1 méga : 900 F.

Pierre-Jean Texier ✆ dom : 93 34 91 50 & bx : 93 95 42 97

---

**Recherche**, pour Apple //e, programme de calculs de résistance de matériaux pour structures métalliques — soft commercial ou développé par un applemaniaque concerné. Recherche également clavier détachable.

Joël Piard : ✆ 90 74 29 64

---

**Vends** Autoworks pour AppleWorks sauf version française 1.4 : 150 F.

Gisèle Perrault ◐ dom : 42 06 55 01 ✆ bx : 42 60 38 64 poste 1149

---

**Vends** Apple //e 65C02 128Ko, moniteur vert, 2 lecteurs 140Ko, ImageWriter I 132 colonnes, carte SuperSérie, Souris.

Marc Goetz - 26 rue du Gal Rampont - 67170 Brumath ✆ soir : 88 53 44 44.

---

**Vends** Imprimante Seïkosha GP 550A + Cordon + interface parallèle Apple : 3000 F.

Jean-François Dupont - 8, rue Madame de Sévigné - 53000 Laval ✆ 43 69 05 77


## Editeur Plein Ecran EPE v 5.1

Le Pacha — Apple ][+, //e, //c, IIGS

- Listez vos programmes Basic en avant et en arrière.
- Modifiez, insérez, effacez des caractères en plein écran sans relire les lignes.
- Recherchez toute chaîne de caractères.
- Choisissez vous-même les codes de contrôle d'EPE.
- Modifiez EPE : le fichier source est sur la disquette.

**200,00 F TTC franco**
**(bon de commande page 71)**


Solution du n° 34

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| E | T | R | I | V | I | E | R | E | S |
| T | O | U | R | A | N | G | E | A | U |
| R | U | G | I | S | S | A | N | T | E |
| I | R | I | S | E | E | ■ | C | ■ | R |
| V | A | S | E | ■ | C | I | A | O | ■ |
| I | N | S | E | C | U | R | I | T | E |
| E | G | A | ■ | I | R | I | S | E | S |
| R | E | N | C | A | I | S | S | E | S |
| E | A | T | ■ | O | T | E | E | ■ | E |
| S | U | E | R | ■ | E | S | S | E | S |

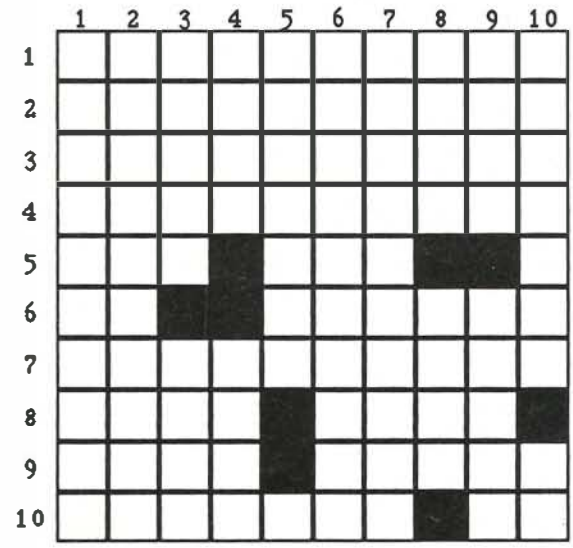
Horizontalement

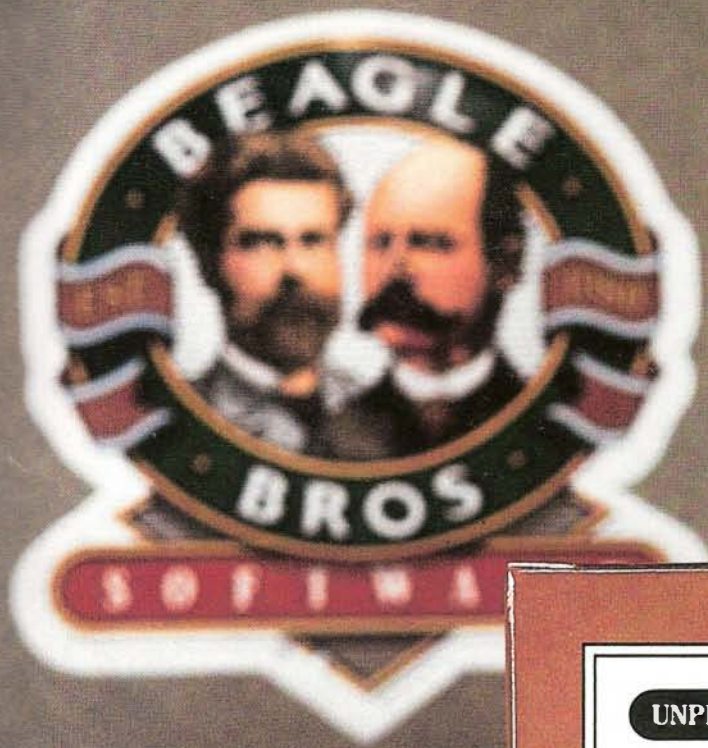
1 - Gendarme
2 - Donc nées
3 - Lézarderont
4 - Dégommante, en quelque sorte
5 - Tutti - Arrivés de travers !
6 - Ile - Sèchera
7 - Mènent en bateau
8 - Demi-suicide - Sarte l'associe au 9.2 horizontal
9 - Nom de séñorita - Vide
10 - Morts - Bien trop petite jupe !

Verticalement

1 - Ordre
2 - Manque de compréhension
3 - Prénom - Bordure
4 - Sorte d'aéronef - Ville de Moldavie
5 - N'est pourtant pas sans signification
6 - Préparaient
7 - Bizarrerie
8 - Il vaut mieux qu'il ne soit pas fils de Camé ! - Se promena
9 - Un peu épais - Brament
10 - Feras une action - Pronom

Problème 35
par
Joëlle Piard






# AppleWorks™ décuplé

Automatiser
AppleWorks™
version française 1.4 ?
Utiliser la souris ?

**SuperMacroWorks**
de Randy Brandt

---

Programme américain
sur disquette 800Ko
sur disquette 140Ko

Version francisée
sur disquette 140Ko

Documentation
française
Disquette Bonus
de Dimitri Geystor

Documentation
américaine

---

500,00 F

abonnés à Pom's :
450,00 F

Frais de port 20,00 F
Banc d'essai : Pom's 33
Pom's – 12, rue d'Anjou
78000 Versailles
(1) 39 51 24 43

AN APPLEWORKS ENHANCEMENT

**UNPROTECTED**

Backups may be made using standard copying procedures.

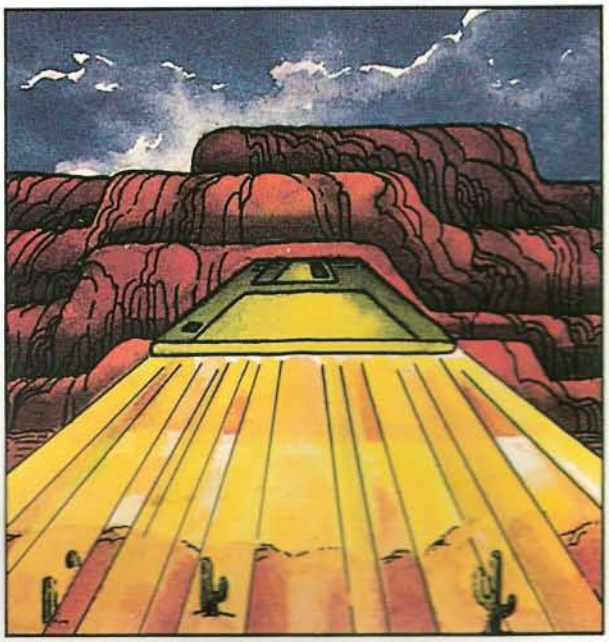
**COMPATIBLE**

Apple IIe, IIc or IIGS
Requires AppleWorks 2.0 or newer

S U P E R

## MACROWORKS

AppleWorks™ Macro Power! Turn any series of keystrokes into a new one-keystroke command. Adds many new features to your Word Processor, Data Base and Spreadsheet.



Certaines nuits ne portent pas conseil...
Pom's
Joker
Hotline Micro
DELAHAYE J. 458.776.059 - Exp 42
Lors qu'au petit matin le problème demeure, il reste la carte Joker. Elle donne un accès privilégié à notre Hotline téléphonique. Gratuite, la carte Joker accompagne chaque abonnement ; un nouvel atout pour les programmes Pom's.